ビュートバランサー2

取扱説明書

ヴイストン株式会社

（ 2015.05.27 ）

目次

1. はじめに

このたびは、弊社製品をお買い上げいただきありがとうございます。

本説明書は、倒立振子学習教材「ビュートバランサー2」（以降、本製品）の使用方法を解説したものです。ご使用になる前に、必ず付属の組み立て説明書および本説明書をご熟読いただき、安全にお使いください。

2. ご注意

- 本製品を取り扱う際には、注意事項に従い正しくお使いください。
- 本製品に強い衝撃を与えないでください。故障する恐れがあります。
- 本製品を水に濡らしたり、湿気やほこりの多い場所で使用したりしないでください。ショートなどによる故障が発生する恐れがあります。
- 本製品から煙が発生した場合、すぐに電源をお切りください。
- 本製品を幼児の近くで使用したり、幼児の手の届くところに保管したりしないでください。
- 動作中、基板上の素子が高温になることがありますので、絶対に触れないでください。
- 基板上の端子（金属部分）に触れると故障する恐れがあります。必ず基板の縁を触るようにしてください。
- 基板上の端子同士が金属などでショートすると、故障する恐れがあります。
- 本書で紹介されるライブラリ・サンプルソースにお客様が変更を加えられた場合、サポート対応の対象外となります。

3. バランサー2 プログラマのインストール

バランサー2 プログラマ（以下、本ソフトウェア）は USB ケーブルで PC と接続した本製品に対して、センサ値の確認・ゲインの調整・ログの取得を行えるソフトウェアです。この章では本ソフトウェアのインストール方法について解説します。

3.1. 動作環境

本ソフトウェアを実行するには以下の動作環境が必要です。

- 対応 OS　　Windows XP SP3 以降 / Vista / 7 / 8 / 8.1（日本語版）
- 解像度　　1024×768 以上・フルカラー表示可能

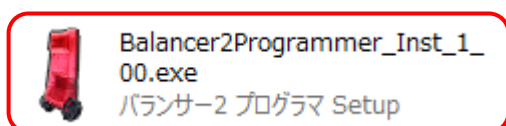
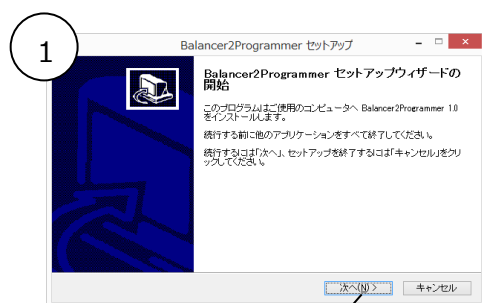
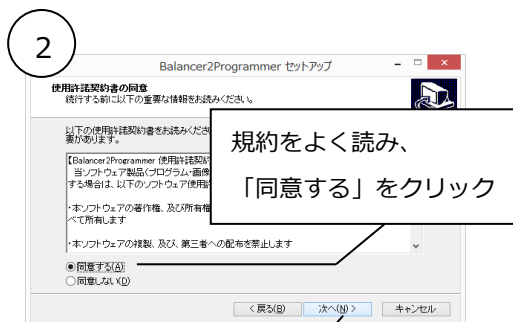
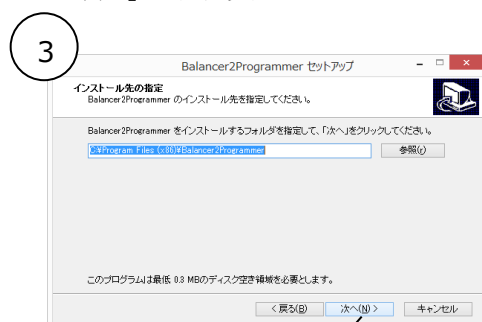
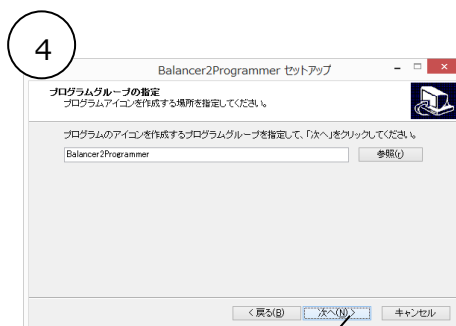
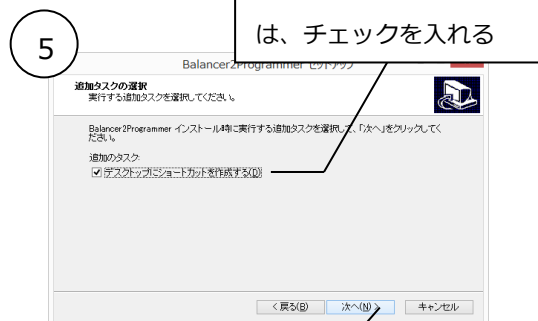
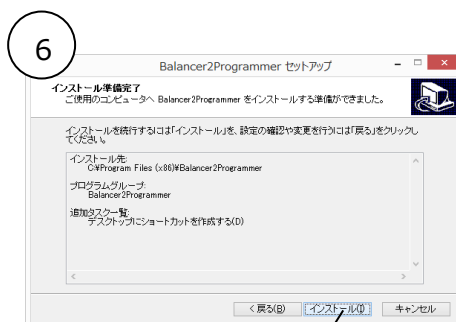
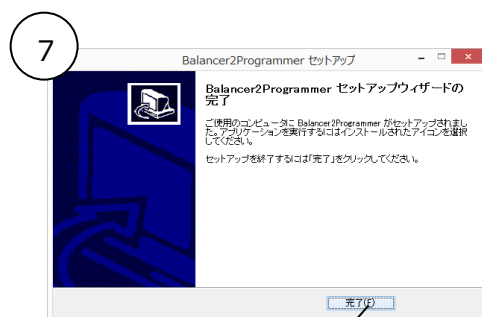
※本製品には、「.Net Framework 4 以上」が必要です。ご使用される PC にインストールされていなければ、Microsoft のホームページよりダウンロードし、インストールしてください。

「.NET Framework 4」 ダウンロードサイト（Microsoft）
http://www.microsoft.com/ja-jp/download/details.aspx?id=17851

3.2. インストール

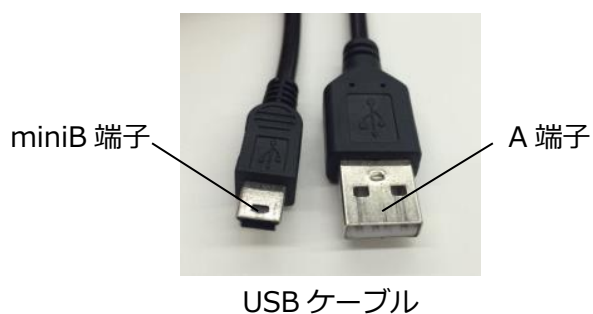
A) 以下の URL からインストーラをダウンロードします。

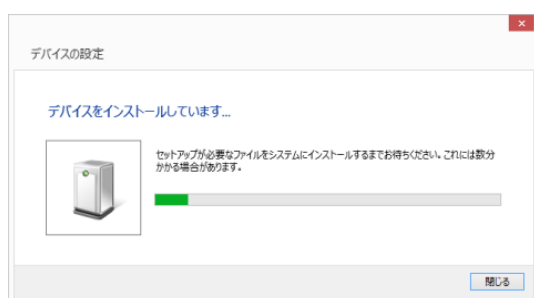
http://www.vstone.co.jp/products/beauto_balancer_2/download.html

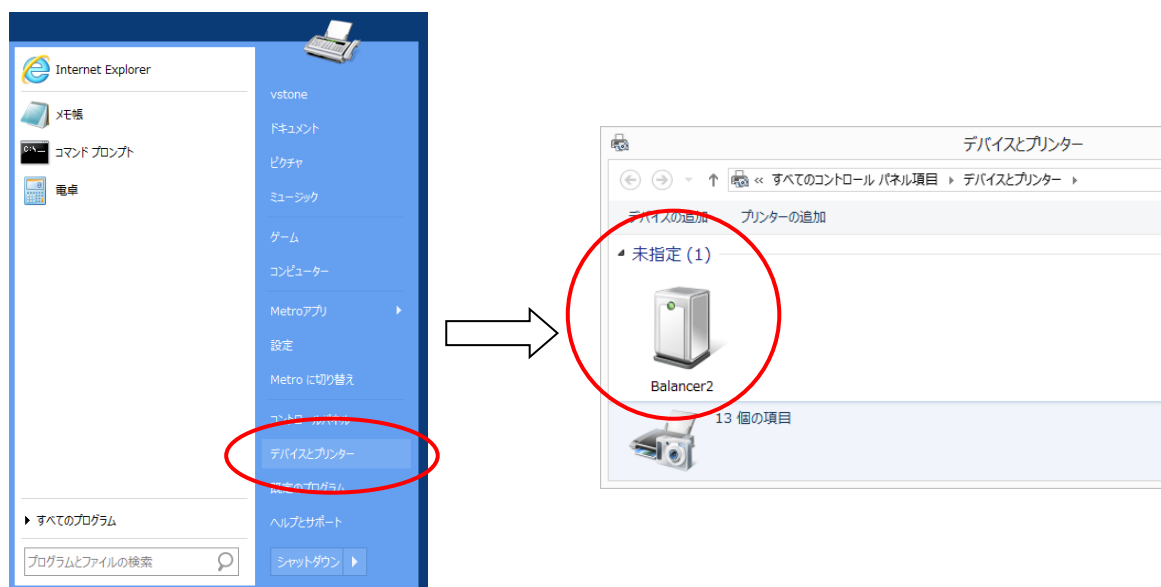
B) ダウンロードが完了したら、zip ファイルを解凍しフォルダ内の「Balancer2Programmer_Inst_***.exe」を実行します（***には、バージョン番号が入ります）。

実行すると以下の画面が表示されます。以下の手順でインストールを行います。

以上でインストールは完了です。デスクトップに作成されたショートカットから本ソフトウェアを起動できます。

3.3. 本体との接続

バランサー2 本体（以下、本体）と PC を USB ケーブルで接続し、PC に本体を認識させると、通信に必要な USB HID が自動的にインストールされます。以下の手順でご確認ください。

A) USB ケーブル（miniB ケーブル）を用意し、PC と本体を接続します。

USB ケーブル　　本体側面に USB(miniB 端子)を接続します。

B) 接続すると以下のウィンドウが表示されますので、しばらく待ちます。

C) 数秒～数分待つと、自動的にインストールが完了し、ウィンドウが閉じられます。

D) ドライバがインストールされているかを確認するには、「デバイスとプリンター」を起動します。

E) 「未指定」の項目の中に、「Balancer2」と表示されていれば正常にインストールされています。

4. バランサー2 プログラマの使用方法

ここでは、使用方法を手順にそって解説します。

4.1. 起動

スタートメニューもしくはデスクトップアイコンから本ソフトウェアを起動します。

4.2. 操作画面

「リスト」
メモリマップの一覧が表示されます

「グラフ」
リストのチェックを入れた項目の「数値」がグラフ化されます

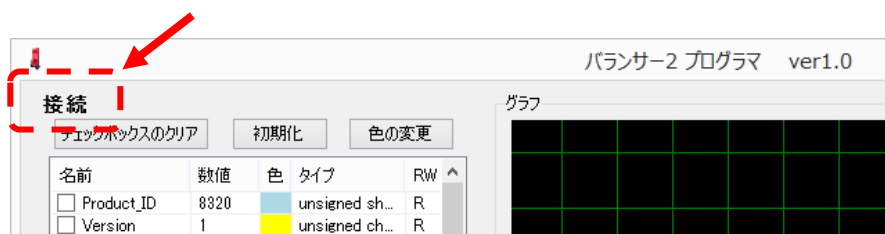
「ログ」
ログを取得するか設定し、取得したログを保存できます

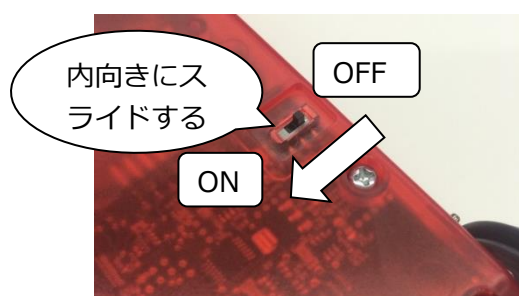
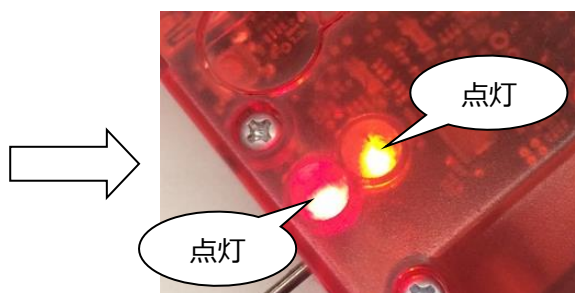
「ゲイン調整」
各ゲインの値を調整し、RAMへ転送することができます

「操縦モード」
矢印ボタンを押すと、バランサー2を操縦することができます

4.3. USB ケーブルの接続

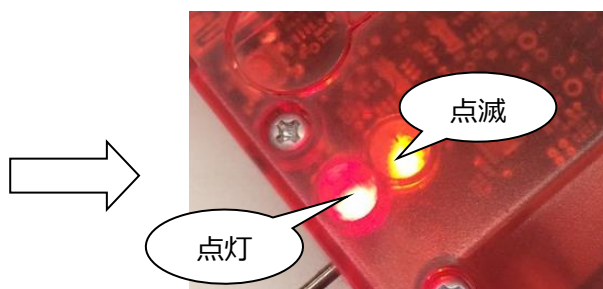
接続方法は、3.3 項「本体との接続」をご参照ください。

バランサー2 の CPU ボードを PC が認識できれば、操作画面の左上に「接続」と表示されます。また、接続できていない場合は、「未接続」と表示されます。

4.4. 電源を入れる

本体のスライドスイッチを内向きにスライドして電源を入れます。電源を入れると、赤色 LED と橙色 LED が両方点灯します。

本体のスライドスイッチを内向きにスライドします。

2 つの LED が点灯します。

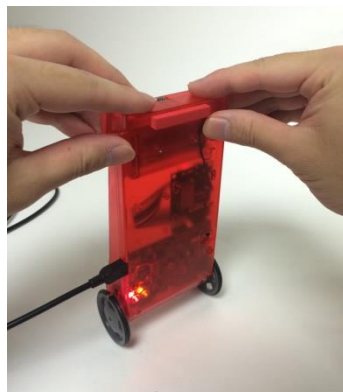
4.5. キャリブレーション

キャリブレーションは、水平な場所に本体を寝かせて行います（ボタン・スライドスイッチは上向き）。本体の操作ボタンを 1 度押すとキャリブレーションが始まり、約 5 秒間赤色 LED が点滅します。キャリブレーションが完了すると、赤色 LED は点灯し、橙色 LED は点滅します。

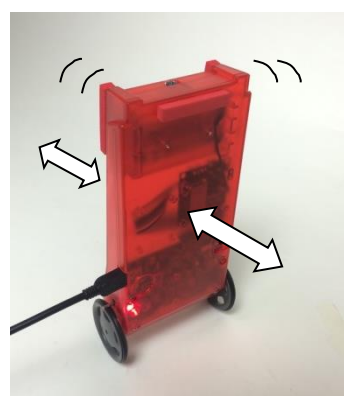
本体の操作ボタンを押すと、赤色 LED が点滅します。

キャリブレーション完了後、赤色 LED が点灯し、橙色 LED が点滅します。

4.6. 倒立動作の開始

本体の倒立動作を開始させます。

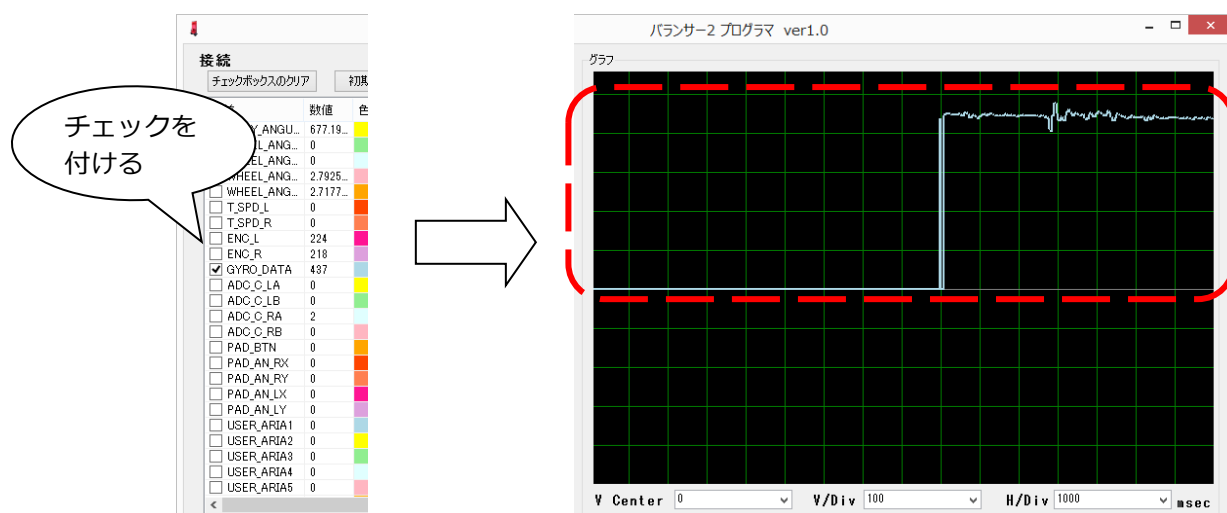
A) 本体を手動で倒立状態の姿勢にして手で支えます。

B) 操作ボタンを押すと、本体が倒立動作を開始します。姿勢を保ったまま操作ボタンを押してから、ゆっくりと手を離して下さい。

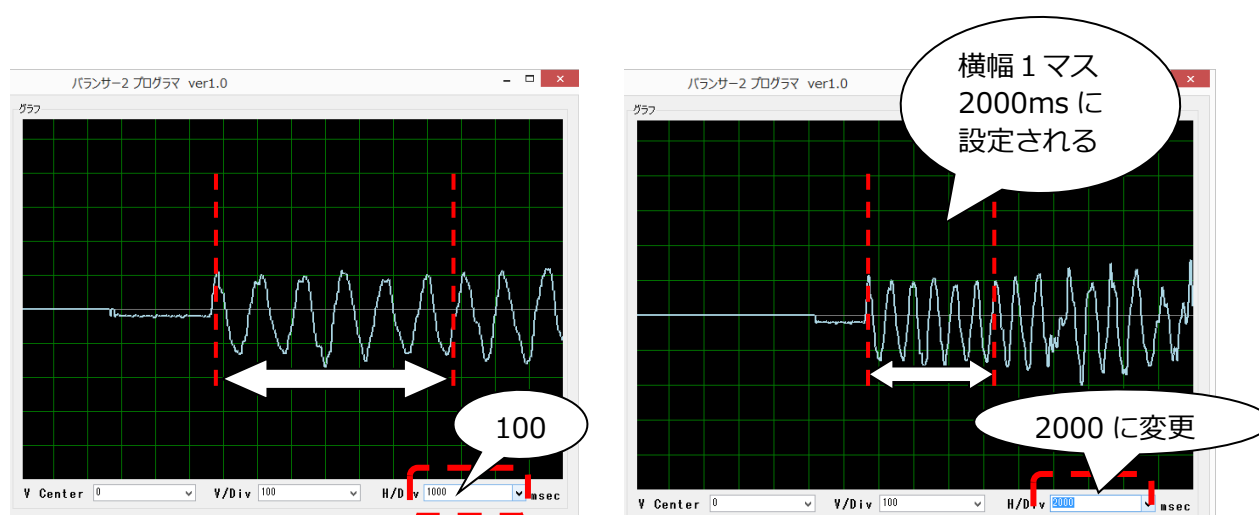
※ボタンを正しく押さないと動作せずに倒れてしまいますので、倒れた場合も手で支えられるように、動作後はゆっくりと手を離してください

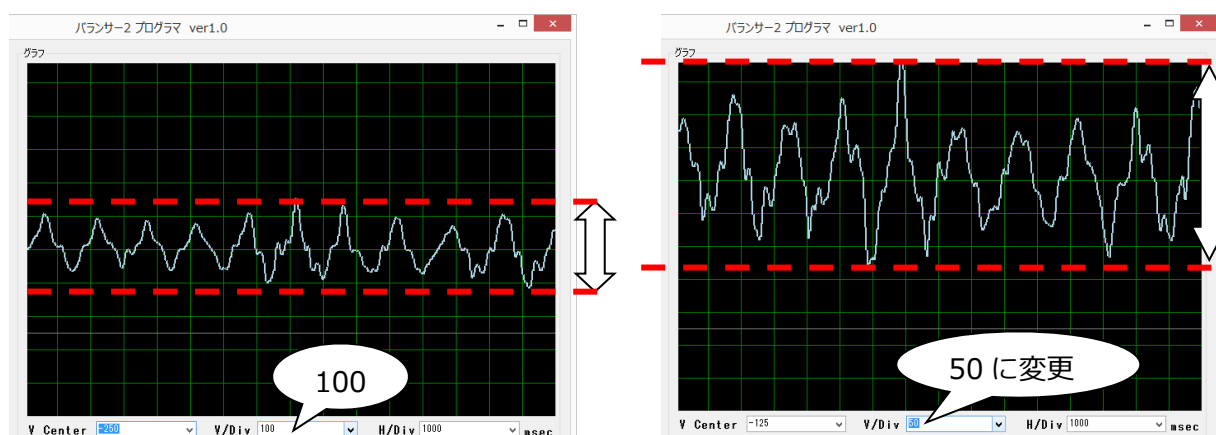
停止中

操作ボタンを押すと、本体が倒立動作を開始します

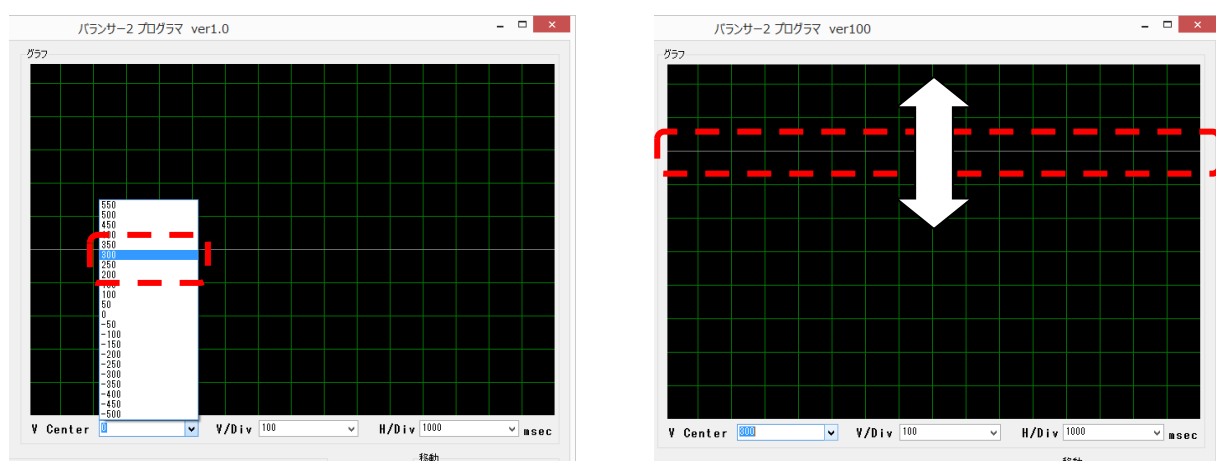
4.7. グラフを確認

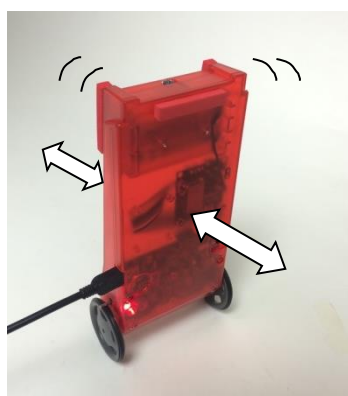
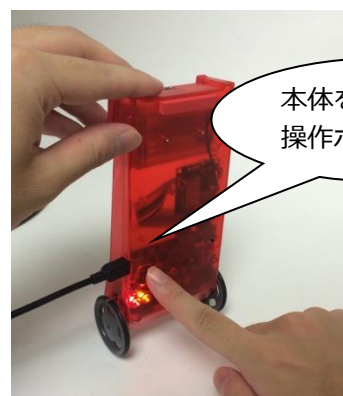
リスト内の表示したいパラメータのチェックボックスをクリックしチェックを付けると、グラフに新しく線が表示されます。チェックした行の項目の「数値」がグラフに反映されます。

※画面に表示されているリストは、11.7 項「ライブラリメモリマップ」と同じ内容となっておりますので、そちらもご参照ください。

4.8. 横幅の変更

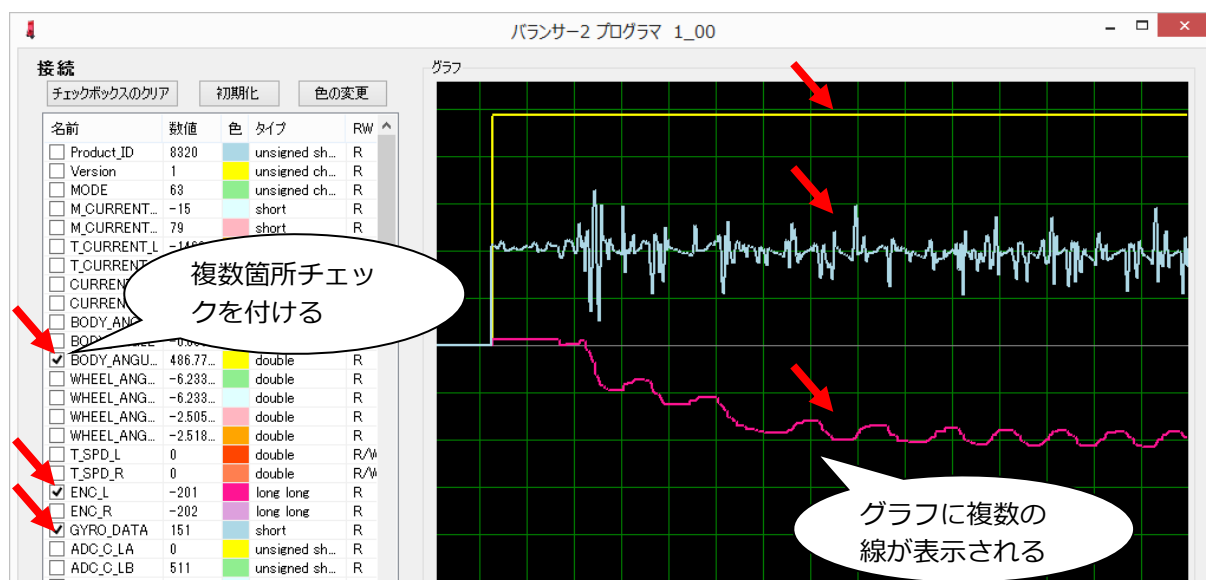
「H/Div」と表示された右隣のコンボボックスの選択内容を変更すると、横軸の幅が変更されます。横軸は時間軸を表します。

4.9. 縦幅の変更

「V/Div」と表示された右隣のコンボボックスの選択内容を変更すると、縦軸の幅が変更されます。縦軸はリスト内のパラメータの軸を表します。

4.10. 中央線の移動

「V Center」と表示された右隣のコンボボックスの選択内容を変更すると、中央線が移動します。

4.11. 停止させる

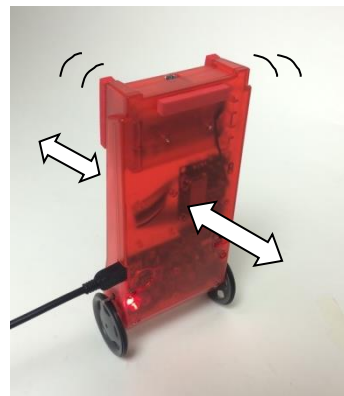
本体を停止させるには、倒立動作中に操作ボタンを押します。動作を止めると倒れてしまうため、<u>必ず、本体を手で支えてから停止させてください</u>。

倒立動作中　　　操作ボタンを押すと、本体が動作を停止します

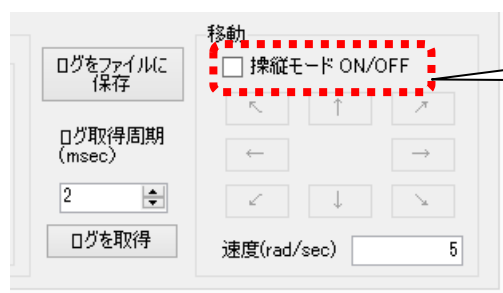
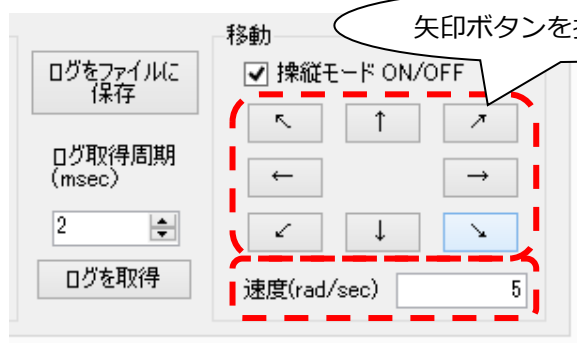
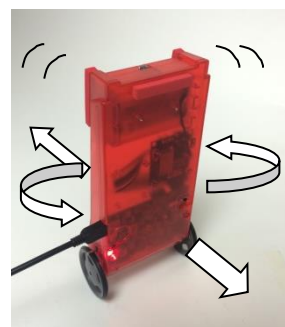
4.12. グラフを追加する

4.7「グラフを確認」と同様に、グラフを追加したいデータのチェックボックスを複数箇所クリックしチェックを付けると、グラフに線が追加されます。

※チェックしたにも関わらずグラフに線が表示されない場合は、4.9 項「縦幅の変更」や、4.10 項「中央線の移動」を参考に、グラフ表示域を調整してください（数値が大きすぎる、あるいは小さすぎることにより、グラフの枠内に線が収まらないことがあります）。

4.13. 操縦モードで操縦する

A) まず、本体と PC を USB ケーブルで接続した状態で、4.6 項「倒立動作の開始」と同様に、倒立動作を開始させます。

停止中

操作ボタンを押すと、本体が倒立動作を開始します

B) 操作モード ON/OFF の左隣にあるチェックボックスをクリックしてチェックを付けます。

C) 各矢印ボタンを押すと、「速度(rad/sec)」の右隣にあるテキストボックスの速度で、本体が矢印の方向に進みます。

速度を設定し、矢印ボタンを押します
(速度「5」の場合 5[rad/sec]の速度で進みます)

矢印ボタンの向きに合わせて進みます

※操縦モードで動かす際は、本体の転倒・転落にご注意ください。また、USB ケーブルの届く範囲内で動かしてください。

4.14. ゲインの変更

ゲインを調整するには、以下の枠で囲まれた 12 個のテキストボックスを編集し、「ゲイン」の各値を設定します。

ここでは、BODY ゲインを書き換えてみます。

以下の［1］となっている箇所に、［1.5］と入力します。

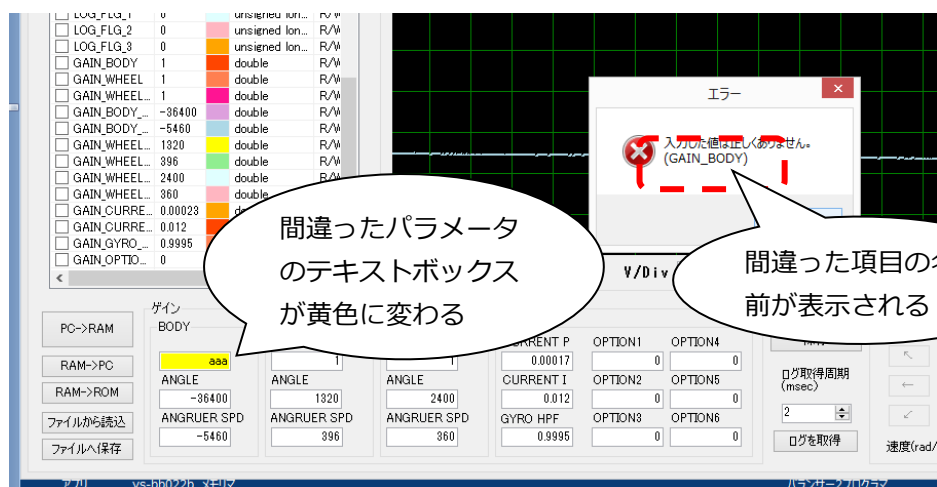
「PC->RAM」ボタンを押すと、バランサー2 の RAM へゲインの各値が送信されます。

この状態でもう一度動作確認してみましょう。

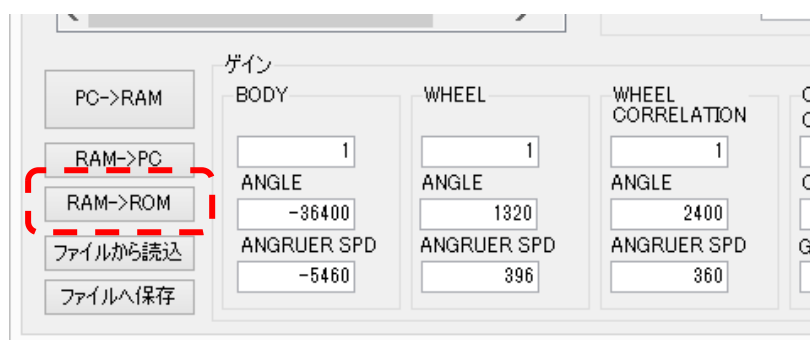
4.13 項「操縦モードで操縦する」と同様に矢印ボタンを押して、本体を動かしてみます。

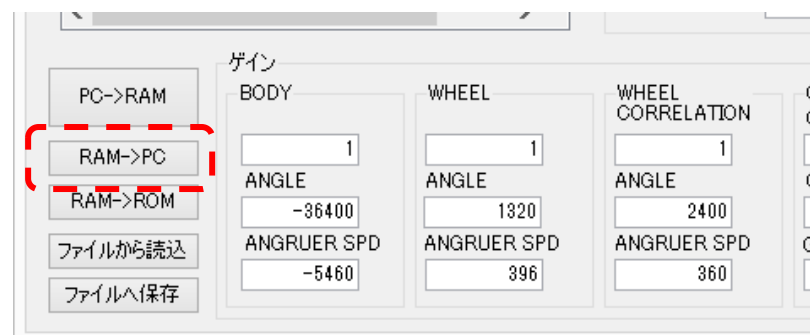
先ほどと動きが変わり、本体の角度・角速度に対するフィードバックが大きくなり、細かい動作をするようになります。このようにゲインの各値を変更することで、倒立動作を調整することができます。

ゲインに関する詳細は、11.9 項「ゲイン・係数」をご参考ください。

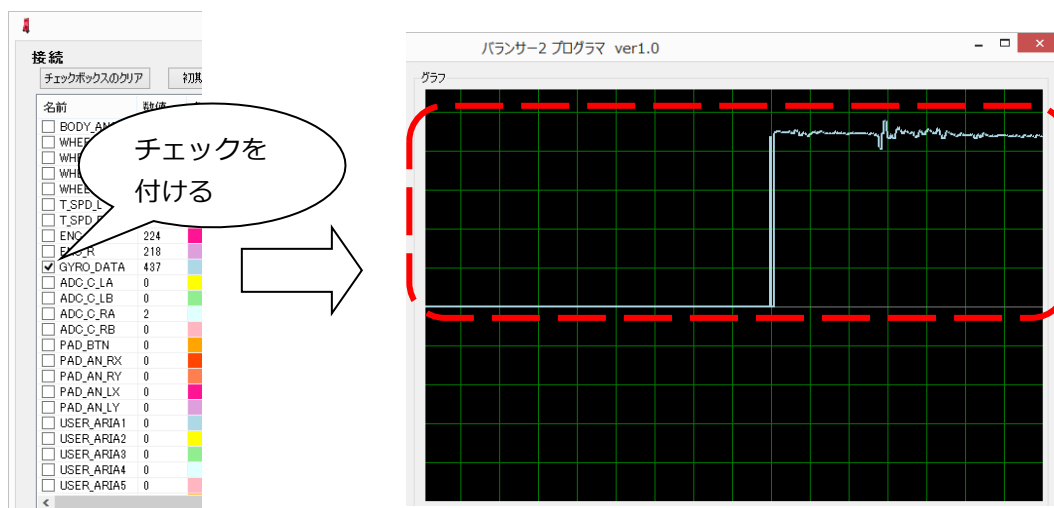
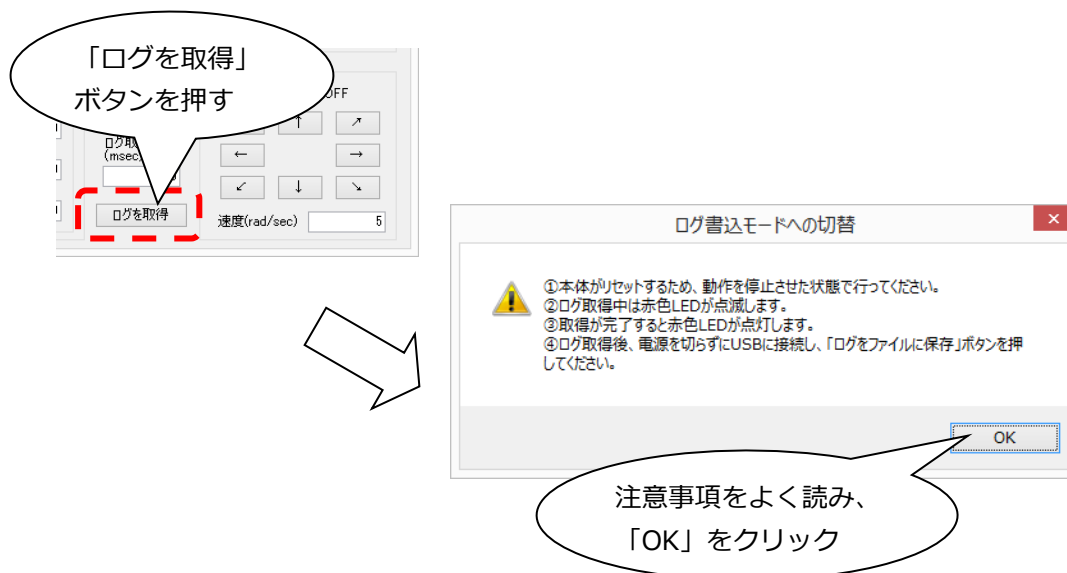
また、ゲインの値を正しく入力せずに「PC->RAM」ボタンを押した場合、エラーメッセージが表示されます。その際、間違ったパラメータの名前が表示されるので、その値を正しく入力し直して、再度「PC->RAM」ボタンを押してください。間違ったパラメータのテキストボックスは黄色に変わります。

4.15. ゲインの書込

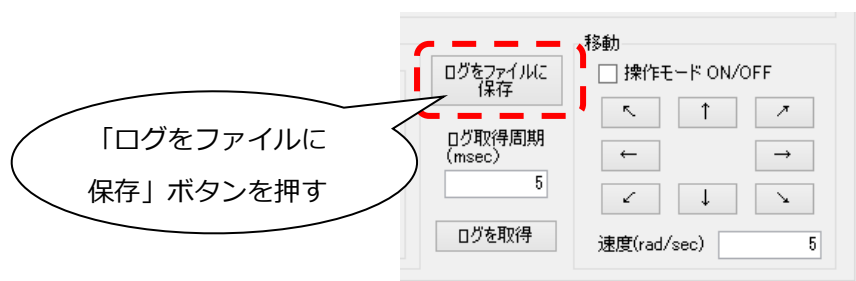
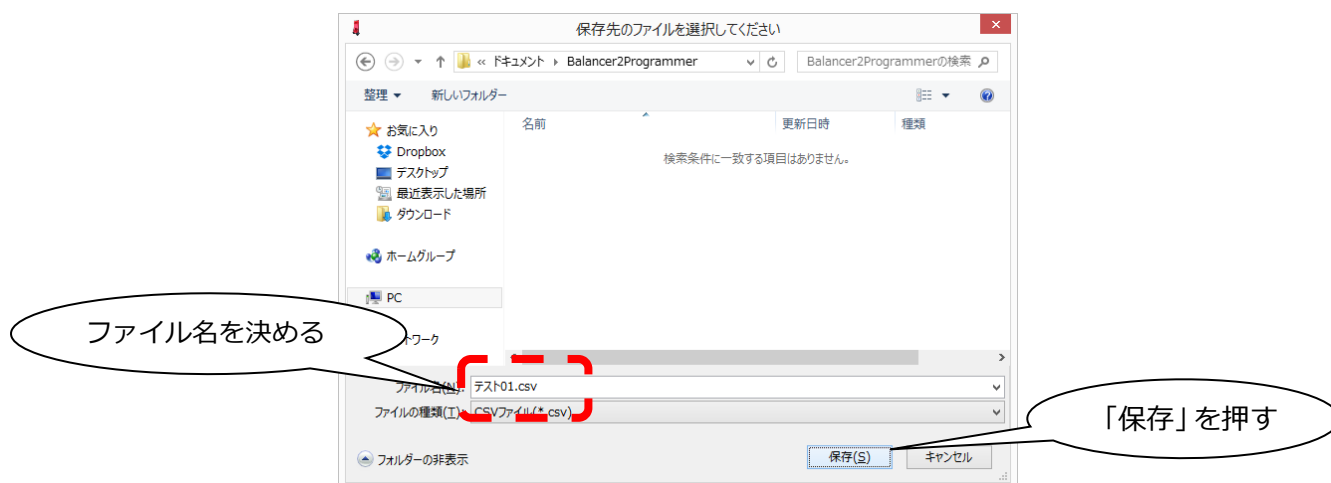
変更したゲインは RAM に送信されただけで、電源を切ると消えてしまいます。

「RAM->ROM」ボタンを押すと、バランサー2 の RAM に送信された設定を FlashROM に書込むことができます。ROM に書き込むことで、電源を切っても、ゲインの値が消えることはなく、次回起動時には ROM に書き込まれたゲインで動作させることができます。

4.16. ゲインの読込

「RAM->PC」ボタンを押すと、バランサー2 の RAM に書き込まれたゲインを、画面内の「ゲイン」の各値が書き込まれたテキストボックスへ反映させることができます。

4.17. ログを取得する

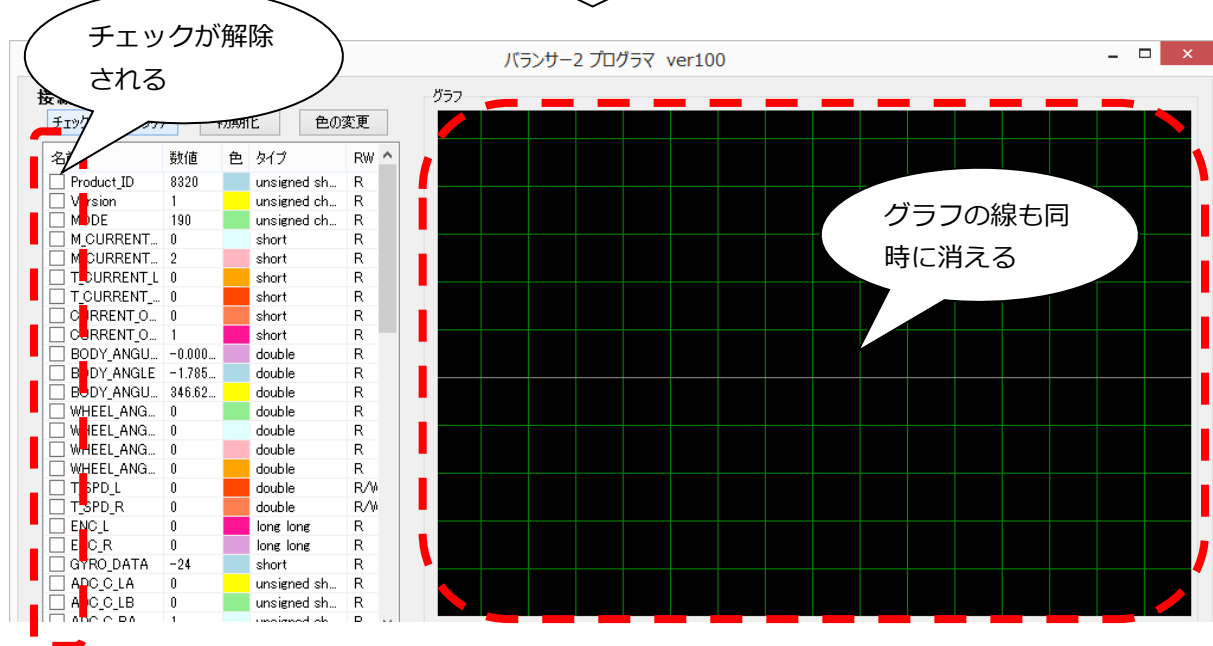
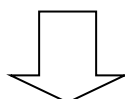
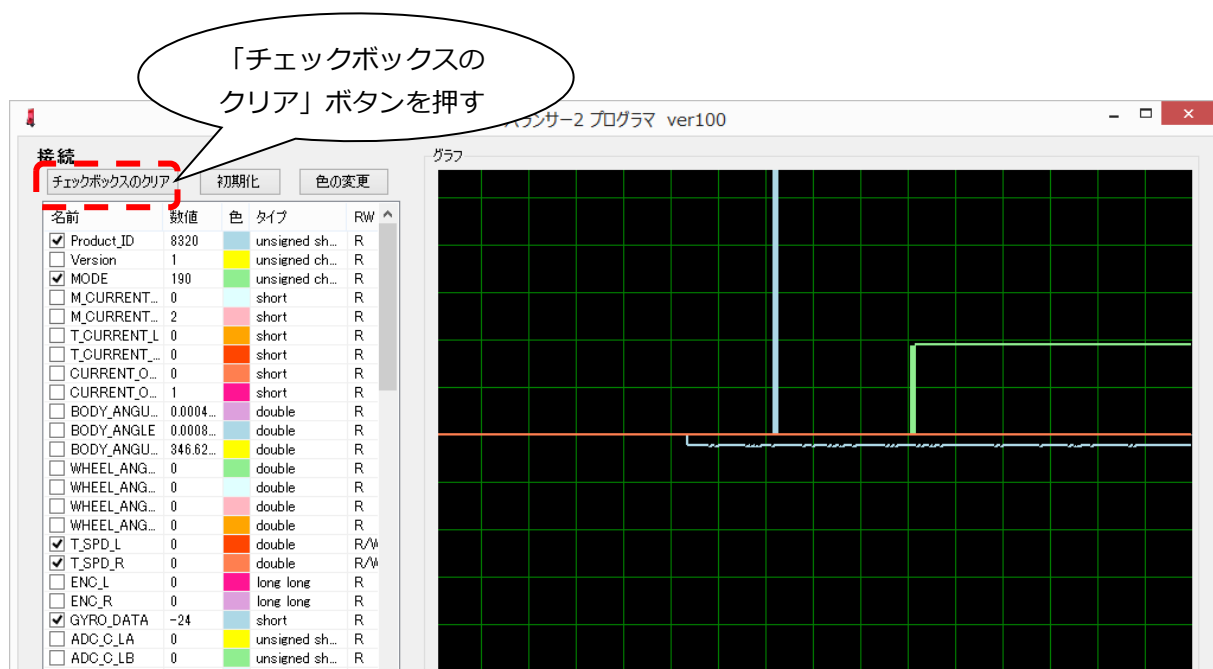
各設定を行うことで、本体の動作中にログを取り、取得したログのデータを PC に保存することができます。本体の RAM 内容量により、合計で 4Kbyte までのログを保存できます。ログを取る項目が多くなったり、各項目のサイズが大きくなったりするとその分ログを取る回数が少なくなります。

A) まず、4.7 項「グラフを確認」と同様に、ログを取得したい項目のチェックボックスをクリックし、チェックを入れます。

B) 次に、ログ書込モードへ切り替えるために、「ログを取得」ボタンを押します。

C)　バランサー2 を動作させ、赤色の LED が点滅から点灯に切り替わるとログの取得が完了します。**そのまま、バランサー2 の電源を切らず**に、PC と接続し、「ログをファイルに保存」ボタンを押します。

※本体の RAM 領域に保存されるため、電源を切るとログデータが消去されます。

D)　保存を行うダイアログが出現します。保存先とファイル名を決め、「保存」ボタンを押すと、ログが保存されます（初期のファイル名は、「新しいファイル.csv」になります）。

5. バランサー2 プログラマの機能

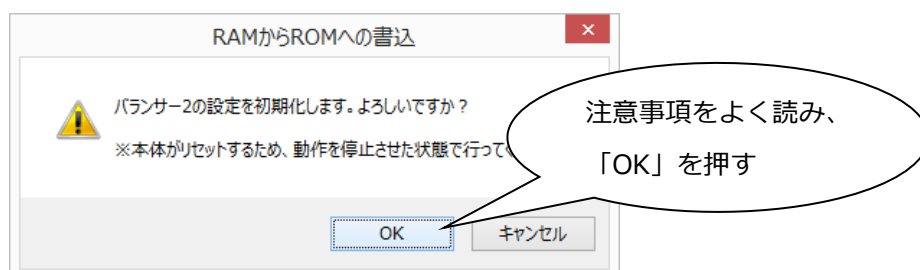
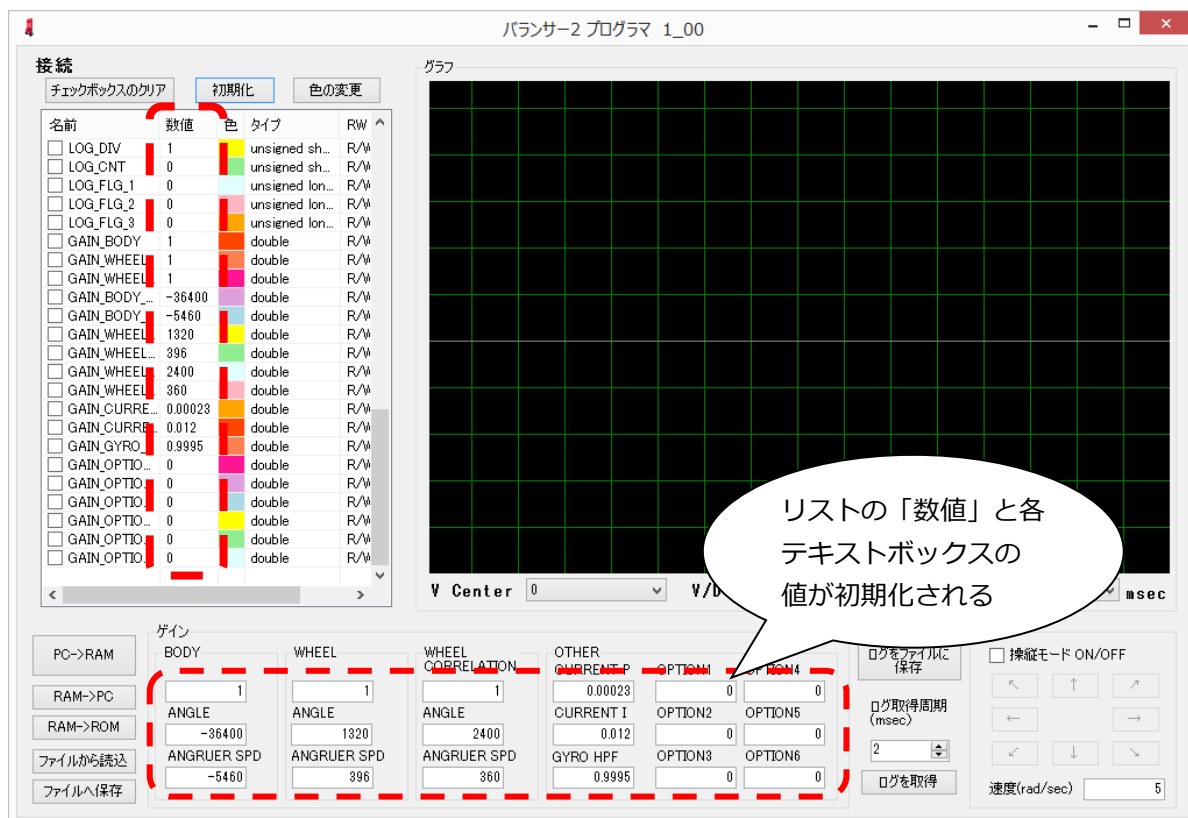
ここではバランサー2 プログラマのその他の機能について解説します。

5.1. チェックボックスのクリア

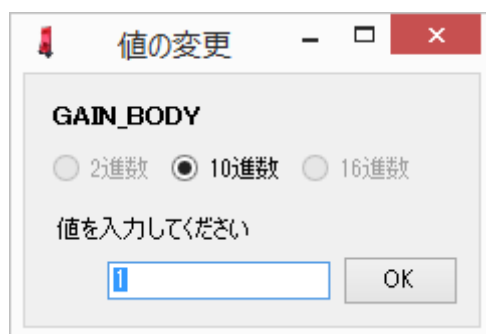
「チェックボックスのクリア」ボタンを押すと、各チェックボックスに付いたチェックが全て解除されます。また、解除と同時にグラフの線が消えます。

5.2. 初期化

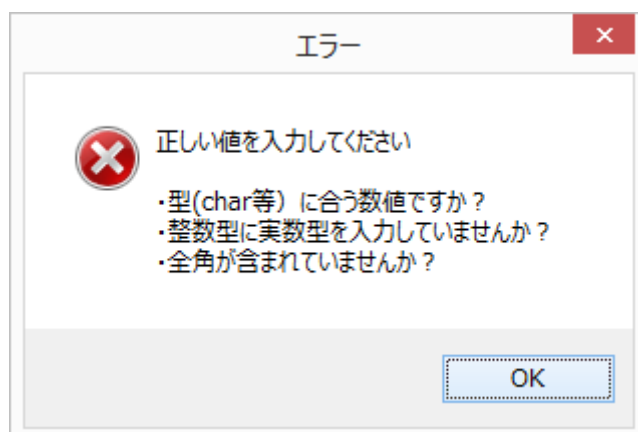
「初期化」ボタンを押すと、本体の ROM の各値がすべて初期状態に上書きされ、ゲインとログ取得周期のテキストボックスも初期状態になります（本体と PC が接続中の状態でのみ動作します）。

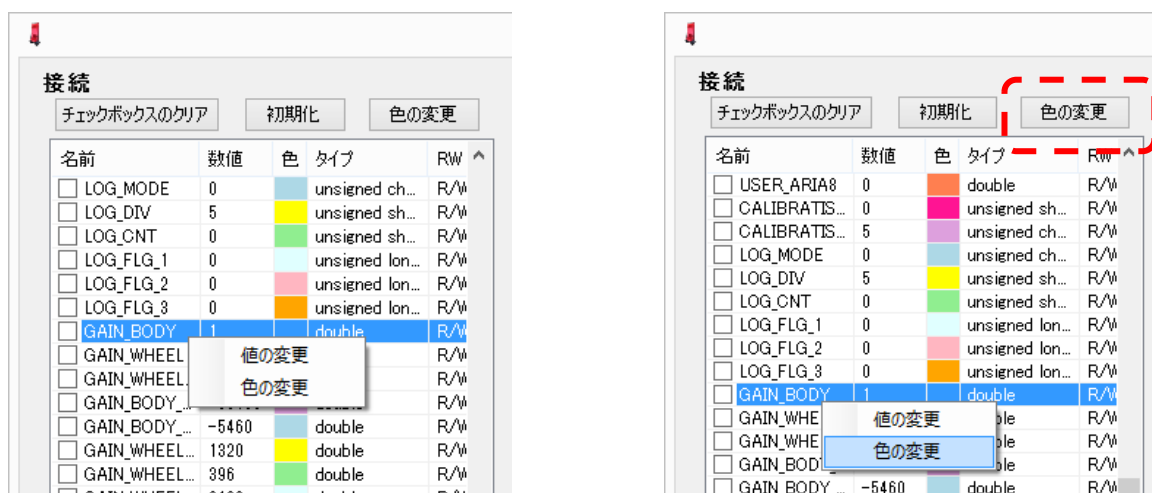
A)　まず、「初期化」ボタンを押します。

B)　初期化を行うか確認するメッセージボックスが表示されるので、注意事項をよく読み、「OK」を押します。

C)　本体の ROM のデータと各テキストボックスの中身が初期化されます。

5.3. 値の変更

リストをマウスで右クリックすると、「値の変更」と「色の変更」の項目が記載されたメニューが表示されます。

「値の変更」では、新しいダイアログにて数値を入力し、指定した項目の値のみ、RAM へ直接書き込むことができます。

※　「double」型は、10 進数の実数のみの入力となり、それ以外の型は 2 進数・10 進数・16 進数の整数のみ入力が可能となります。

※　間違った値を入力して OK を押すと、以下のようなダイアログが表示され、RAM へ書き込むことができない仕様となっています。

5.4. 色の変更

5.3「値の変更」と同様にリストを右クリックし、「色の変更」を押します。「色の変更」では、グラフに表示する線の色を変更できます。この機能は、「色の変更」ボタンと同様の機能を持ちます。

A) 「色の変更」をクリックすると、「色の設定」ダイアログが表示されます。この設定で色を決定し、「OK」をクリックします。

6. 拡張基板の取付

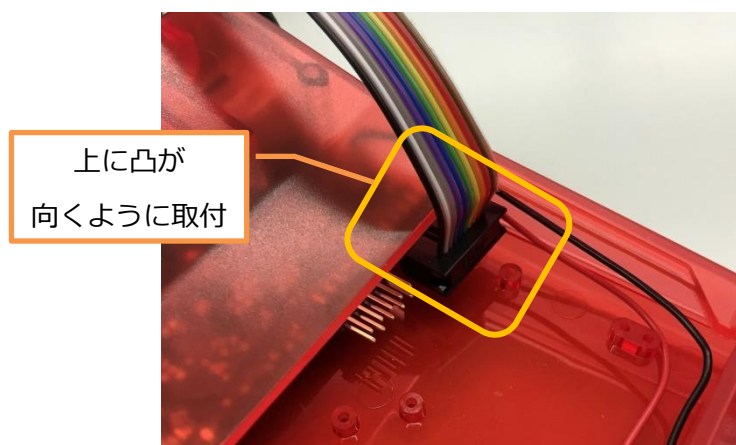
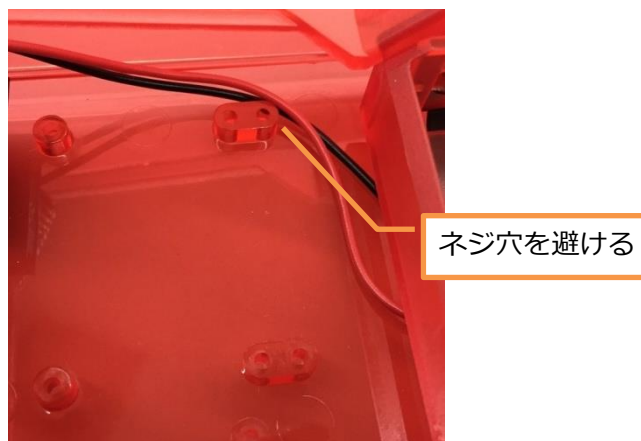
6.1. 無線コントローラ VS-C3 の取付

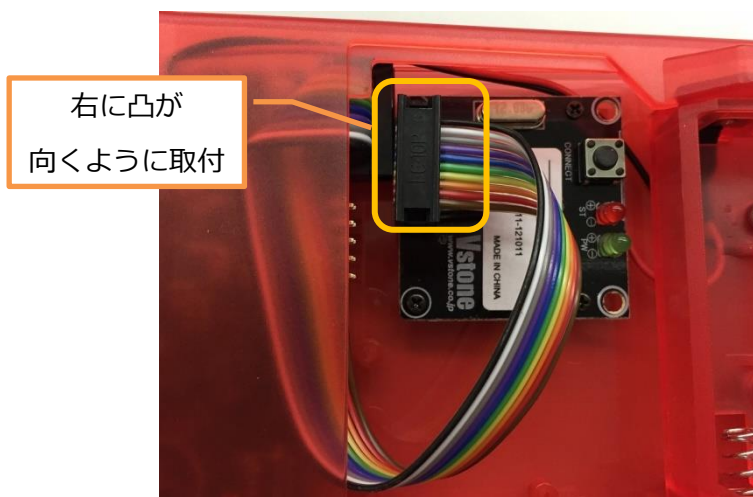
無線コントローラ VS-C3（別売）を取付けると、進行方向・回転をアナログスティックで操縦できるようになります。

※操縦の制御はライブラリ外で行っていますので、出荷時のファームウェアの場合に限ります。
　サンプルを参考に制御プログラムを作成してください。

1)　本製品とは別に、以下の部品を用意します。

無線コントローラ　VS-C3	1 セット
10 芯コネクタ＋フラットケーブル（圧着済み）　150mm	1 個
ねじ D（M2-3 低頭タッピンねじ）	4 本

2)　フラットケーブルを本体基板の CN10 に、コネクタの凸が上（電池蓋開口部）を向くように取り付けます。

3)　噛み込まないように、電源ケーブルをネジ穴の外側に避けます。

4) 本体右側の取り付け穴に、VS-C3 の受信機（VS-RCV3）を以下の向きにネジ D で固定します。

5) フラットケーブルを VS-C3 の受信機に、コネクタの凸が基板の内側を向くように取り付けます。

6) ケーブルを蓋が閉まるように整えます。

7) VS-C3 と本製品に電池を入れ、いずれもスイッチを入れます。

8) VS-C3 の取扱説明書を参考に、ペアリング操作をします。

9) ペアリングが完了したら、蓋を閉めます。

10) 倒立制御をさせ、動作状態にしたあと、左アナログスティックの前後で前進・後退、右アナログスティックの左右で左旋回・右旋回を操縦できます。

6.2. Bluetooth モジュール VS-BT001 の取付

VS-BT001（別売）を取り付けると、無線でデータの取得や本体の移動制御を、Bluetooth を搭載したPC、Android スマートフォンなどから行うことができます。

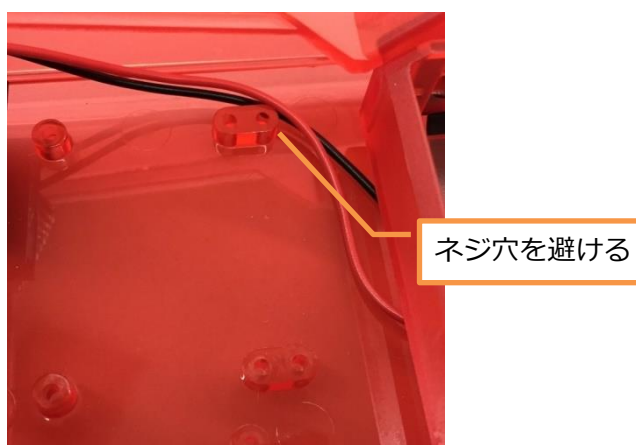
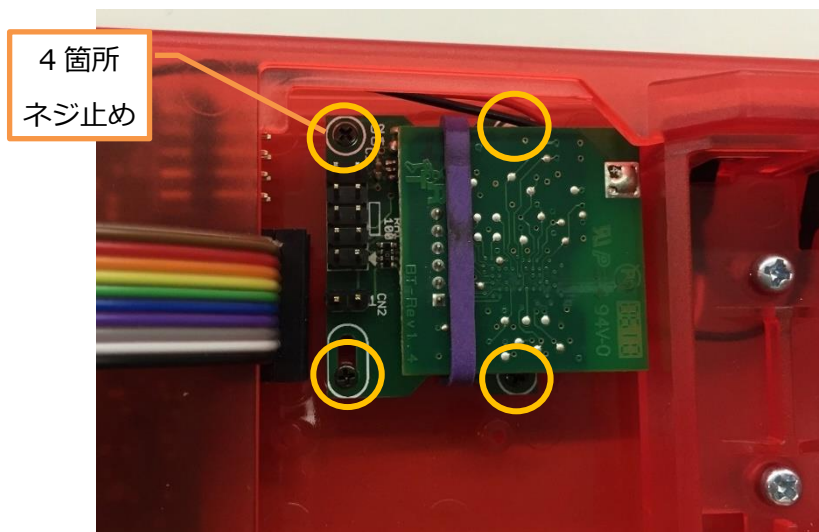
制御プログラムのサンプルソース［Balancer2_firm_Uart］に UART を用いた通信に関する処理が含まれています。

※Android スマートフォン用アプリ　VS-C2 には対応しておりません。

1) 本製品とは別に、以下の部品を用意します。

Bluetooth モジュール VS-BT001		1 セット
10 芯コネクタ＋フラットケーブル（圧着済み）	150mm	1 個
ねじ D（M2-3 低頭タッピンねじ）		4 本

2) フラットケーブルを基板の CN9 に取り付けます。コネクタの凸が上（電池蓋開口部）を向くように取り付けます。

3) 噛み込まないように、電源ケーブルをネジ穴の外側に避けます。

4)　本体右側の取り付け穴に、VS-BT001 を以下の向きにネジ D で固定します。

5)　フラットケーブルを VS-BT001 に、コネクタの凸が基板の内側を向くように取り付けます。

6)　ケーブルを蓋が閉まるように整えます。

7)　取り付けたあと、本製品の電源を入れ、VS-BT001 の内側の緑色の LED が点灯していることを確認して下さい。

8)　VS-BT001 の取扱説明書を参考に、PC、Android スマートフォンなどとペアリングし、ターミナルソフトウェアなどで通信を確認して下さい。

6.3. アナログ入力拡張ボード VS-IX008 の取付

VS-IX008（別売）を取付けるとアナログ出力のセンサを拡張し、制御プログラムにて使用することができます。

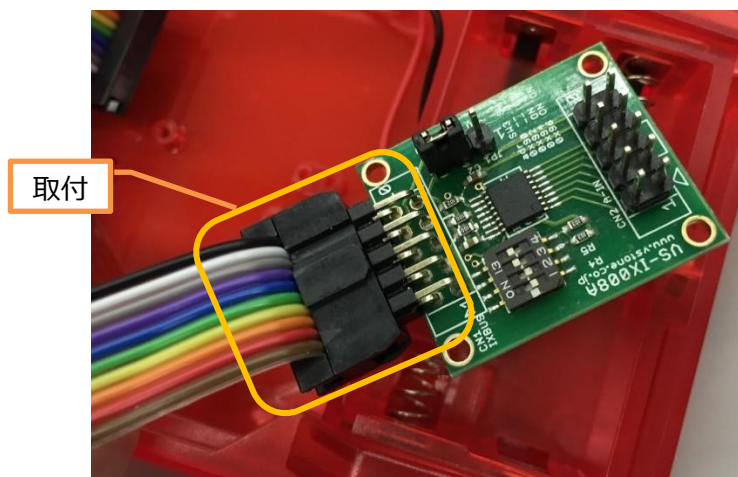
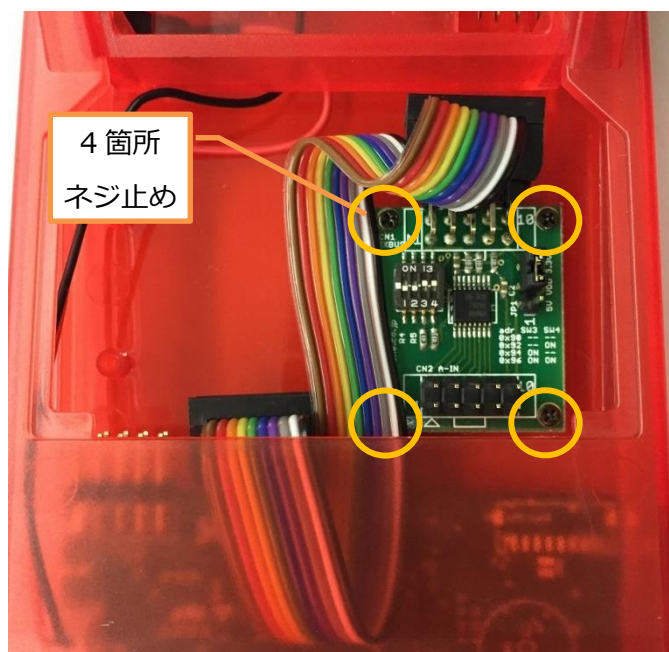
制御プログラムを作成する場合、サンプルソース［Balancer2_firm_ADC_IX008］をご参考ください。

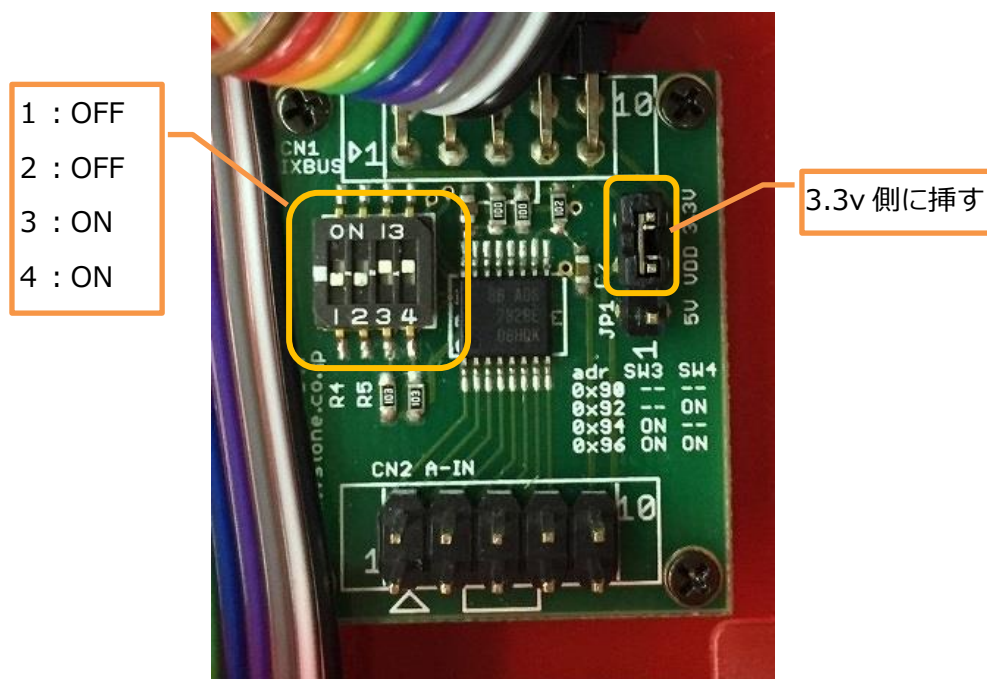
1) 本製品とは別に、以下の部品を用意します。

アナログ入力拡張ボード VS-IX008	1 セット
ねじ D（M2-3 低頭タッピンねじ）	4 本

※VS-IX008 付属のフラットケーブルはコネクタが圧着されていません。圧着が難しい場合、別途圧着済みのケーブルをお買い求めください。

2) フラットケーブルを基板の CN9 に取り付けます。コネクタの凸が上（電池蓋開口部）を向くように取り付けます。

3) フラットケーブルを VS-IX008 に、コネクタの凸が基板の上側を向くように取り付けます。

4) 本体右側の取り付け穴に、VS-IX008 を以下の向きにネジ D で固定します。

5) ケーブルを蓋が閉まるように整えます。

6) VS-IX008 のジャンパピンを JP1 の 3.3v 側に差し込み、DIP スイッチを以下のように設定します。
※サンプルライブラリを使用する場合、VS-IX008 の I2C アドレスを 0x96 に設定する必要があるため

7) センサの接続に関しては VS-IX008 の取扱説明書をご参考ください。

7. C 言語開発環境のインストール

本章では、「ビュートバランサー2」に搭載される MCU「LPC1343」にて実行される制御プログラムの開発方法について解説します。

無償配布されている C 言語統合開発環境「LPCXpresso」を用いたサンプルプログラム・ライブラリを使用します。

「LPCXpresso」評価版のご利用は無料ですが、ダウンロードの際にユーザーカウントを作成する必要があります。また、ビルドサイズの制限がかかっていますので、インストール後にネットワークを利用した認証作業をする必要があります。

7.1. LPCXpresso　ダウンロード

まず、統合開発環境の LPCXpresso をダウンロードします。

1) 以下のサイトにアクセスします。

http://www.lpcware.com/lpcxpresso/home

2) Download the LPCXpresso IDE を選択します。

3)　Windows　を選択します。

4)　Current release の Download LPCXpresso v7.*.* for Windows (current version) を選択するとインストーラのダウンロードが始まります。

※ライブラリ・サンプルソースは v7.3.0 で作成しています。

新しいバージョンで正しく動作しない場合、v7.3.0 をインストールし正しく動作するかご確認ください。

7.2. LPCware へのユーザー登録

ビルドサイズ制限解除のためにはユーザ登録が必要になります。ダウンロードを待っている間に、ユーザ登録を行います。

1） 再度、以下のサイトにアクセスします。

http://www.lpcware.com/lpcxpresso/home

2） Create new account を選択します。

3) 必須事項を記載し、Create new account を選択します。

User account

Home » User account

Create new account | Log in | Request new password

IMPORTANT: PLEASE READ: You must provide a valid email address to use this site. If you don't get the verification email - usually within a few minutes, check your spam box. If you do not get the verification email within 24 hours, please use the FEEDBACK tab to send us an email so we can authenticate your account. Once you register, a verification email will be sent to you with a temporary password to sign onto the site and authenticate your account.

Account information

Username: *

username

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

E-mail address: *

A valid e-mail address. All e-mails from the system will be sent to this address. The e-mail address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by e-mail.

XML sitemap

Personal Information

Full name: *

Optional: Enter your full name. This will not appear on any public pages. The content of this field is kept private and will not be shown publicly.

Company:

Enter your company name. This will not appear on any public pages. Please do not enter a fake name such as 'abc123' as this will cause your account to become blocked. Leave the field blank if you would prefer not to enter your company name. The content of this field is kept private and will not be shown publicly.

Company website:

Enter your company company website URL. This will not appear on any public pages. The content of this field is kept private and will not be shown publicly.

Newsletters

Select the newsletter(s) to which you wish to subscribe.

☐ www.lpcware.com newsletter

CAPTCHA

This question is for testing whether you are a human visitor and to prevent automated spam submissions.

Bd3C2

What code is in the image?: *

Enter the characters shown in the image.

Create new account

選択

4) LPCware から認証メールが送られてきます。そこに記載されている URL をクリックし、ログインします。

これで、ユーザ登録は完了です。

※メールがとどかない場合、再度はじめから行ってください。

7.3. LPCXpresso　のインストール

ダンロードしたインストーラから LPCXpresso をインストールします。

1）ダウンロードしたインストーラを実行します。

2）以下の手順に従って、インストールを進めます。

Setup - LPCXpresso
Debug driver selection
Please select which drivers to install.
Windows may issue warnings when installing these drivers.
If prompted, please allow the drivers to install.
Drivers are located in the "Drivers" directory of the installation
NXP Debug drivers
選択
v7.5.0_254
http://www.nxp.com/lpcxpresso
< Back
Next >
Cancel
Setup - LPCXpresso
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
LPCXpresso v7.5.0_254
Browse...
選択
v7.5.0_254
http://www.nxp.com/lpcxpresso
< Back
Next >
Cancel
Setup - LPCXpresso
Ready to Install
Setup is now ready to begin installing LPCXpresso on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:¥nxp¥LPCXpresso_7.5.0_254
Start Menu folder:
LPCXpresso v7.5.0_254
Additional tasks:
Additional icons:
Create a desktop icon
For all users
Install drivers:
NXP Debug drivers
選択
v7.5.0_254
http://www.nxp.com/lpcxpresso
< Back
Install
Cancel
Setup - LPCXpresso
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
LPCXpresso v7.5.0_254
Browse...
選択
v7.5.0_254
http://www.nxp.com/lpcxpresso
< Back
Next >
Cancel
Setup - LPCXpresso
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing LPCXpresso, then click Next.
Additional icons:
Create a desktop icon
For all users
For current user only
選択
v7.5.0_254
http://www.nxp.com/lpcxpresso
< Back
Next >
Cancel
Setup - LPCXpresso
Installing
Please wait while Setup installs LPCXpresso on your computer.
Extracting files...
C:¥...¥lpcopen_examples_windows_usb_drivers_v1.20¥x86¥WdfCoInstaller01011.dll
v7.5.0_254
http://www.nxp.com/lpcxpresso
Cancel

Setup - LPCXpresso

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

LPCXpresso 7

== Supported Hosts ==
* Following the discontinuation of support by Microsoft, Windows XP is no longer an officially supported platform. LPCXpresso may continue to work on Windows XP but this can no longer be guaranteed. LPCXpresso is no longer tested on Windows XP.
* Mac OS X 10.7 (Lion) is no longer an officially supported platform. LPCXpresso may continue to work on Mac OS X 10.7, but this can no longer be guaranteed. LPCXpresso is no longer tested on Mac OS X 10.7.

== Known Issues ==
* On Windows 8 and 8.1, if both ... and a debug probe that uses the 'Redlink' driver are co... as LPC-Link2 or a

v7.5.0_254

http://www.nxp.com/lpcxpresso

選択

Next >

Setup - LPCXpresso

Completing the LPCXpresso Setup Wizard

Setup has finished installing LPCXpresso on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Launch LPCXpresso v7.5

☑ Display LPCXpresso v7.5 Documentation

☑ LPCXpresso User Guide (PDF)

選択

http://www.nxp.com/lpcxpresso

< Back　Finish

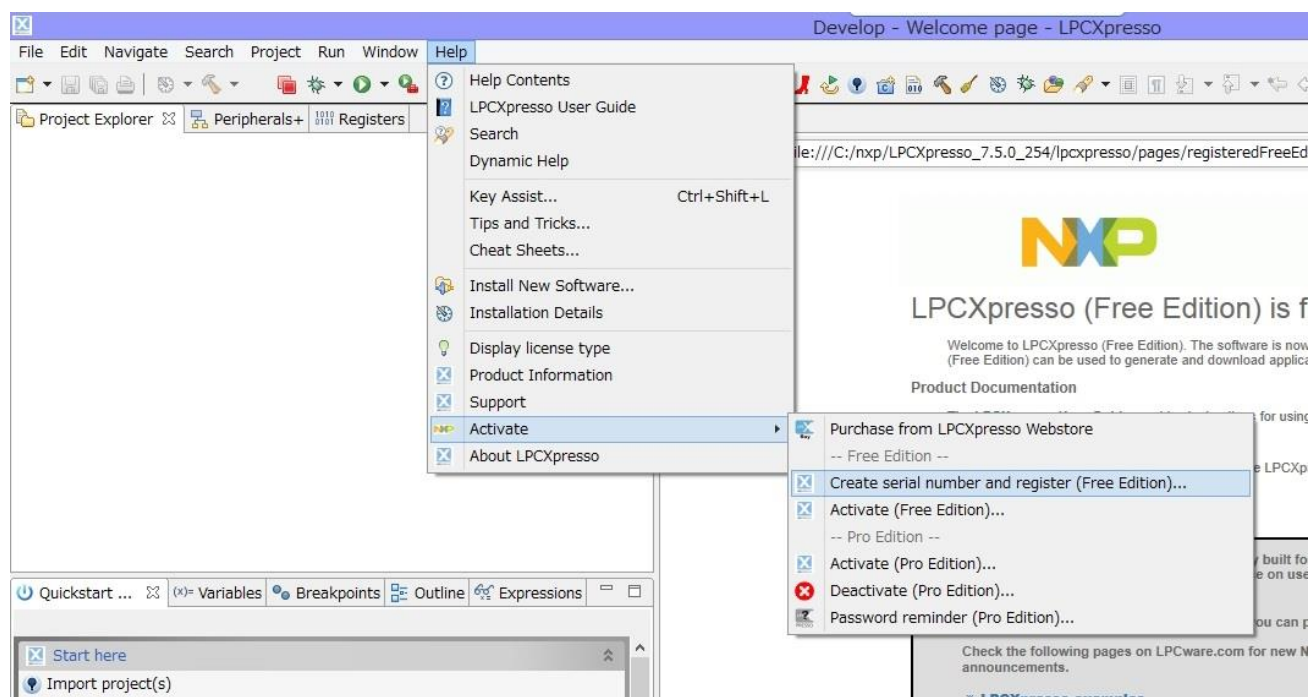
3)　Finish を選択すると、LPCXpresso が起動します。

以上で、LPCXpresso のインストールは完了です。

7.4. LPCXpresso　の起動とアクティベーション

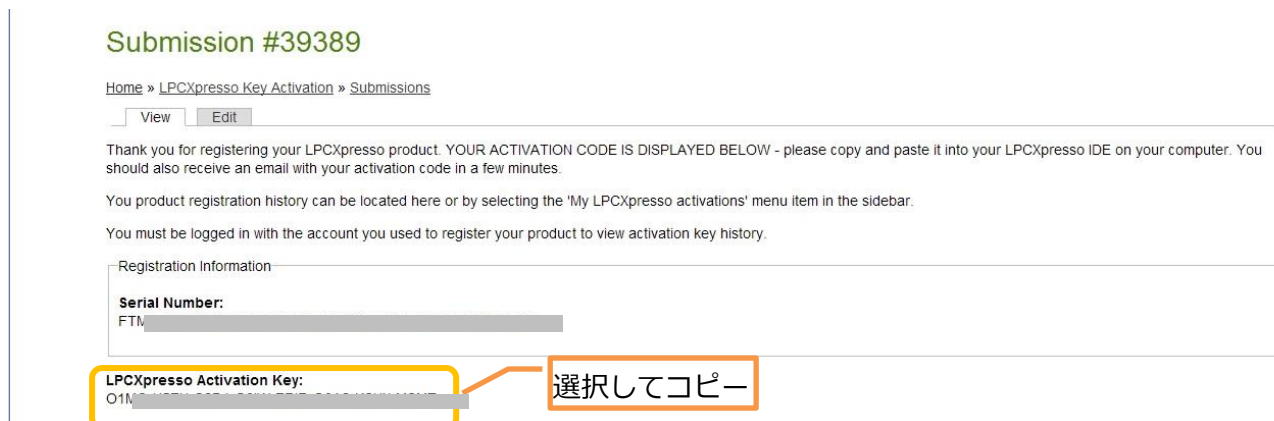
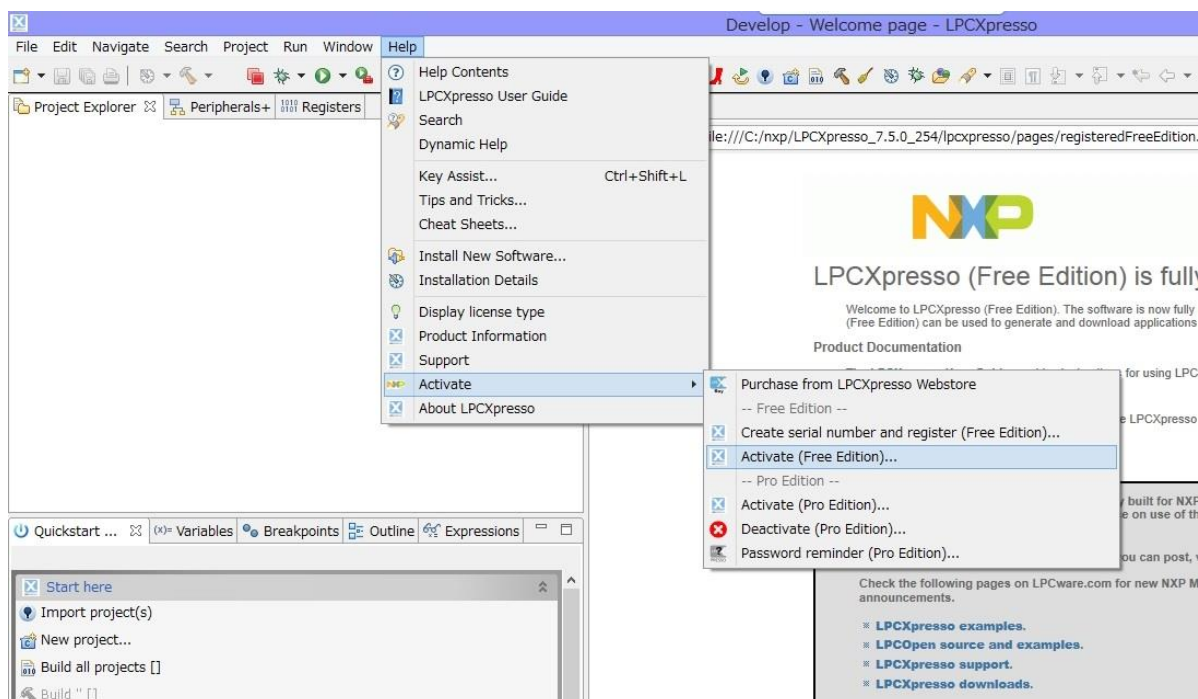
フリー版はビルドサイズに制限がありますので、ライブラリを用いて実行ファイルを作成する場合、認証が必要になります。

認証を行うと 256kbyte までの実行ファイルをビルドできるようになります。

1) LPCXpresso のメニューから[Help] >[Activate]>[Create serial numver and register(Free Edition…]を選択します。

2) Yes を選択します。

3)　[Open in external browser]にチェックを入れ、OK を選択します。

4)　Serial Number が入力されているのを確認し、[Register LPCXpresso]を選択します。

5)　表示された Activation Key をコピーします。

6)　LPCXpresso のメニューから[Help] >[Activate]>[Activate（Free Edition）]を選択します。

7)　コピーした Activation code を貼り付け、OK を選択します。

8)　Yes を選択し、LPCXpresso を再起動します。

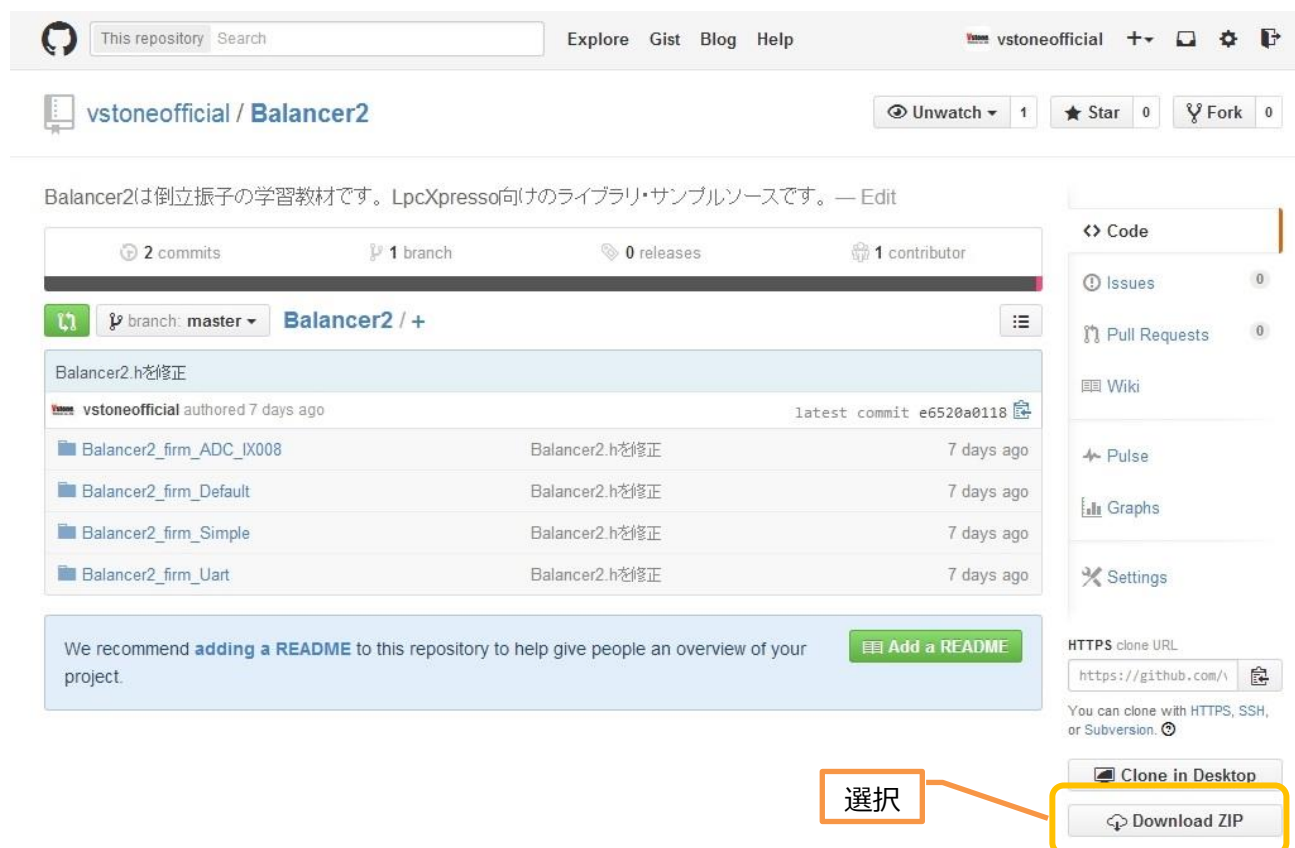
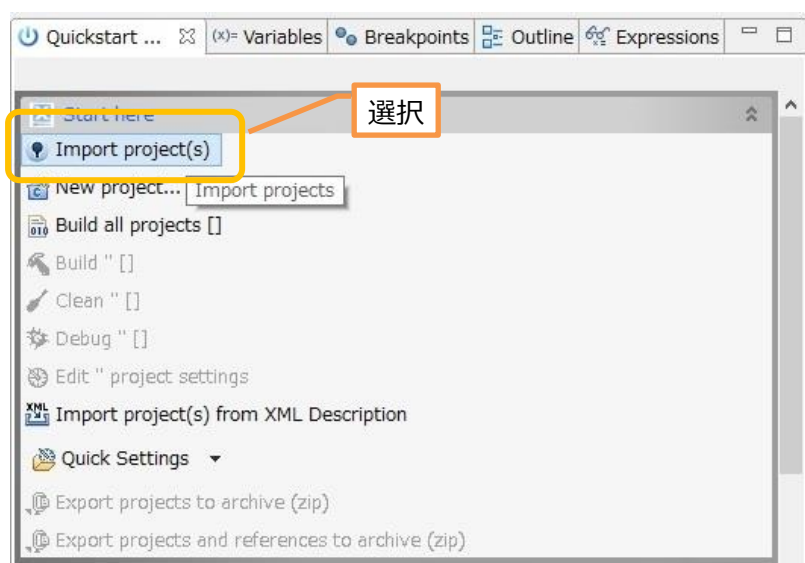
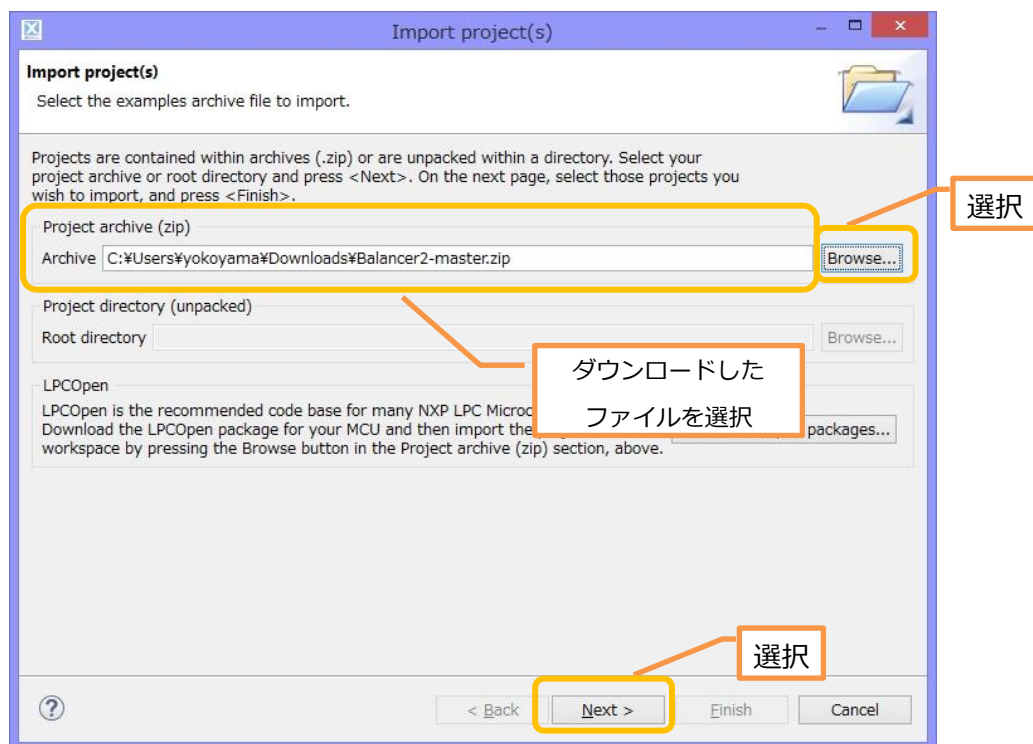
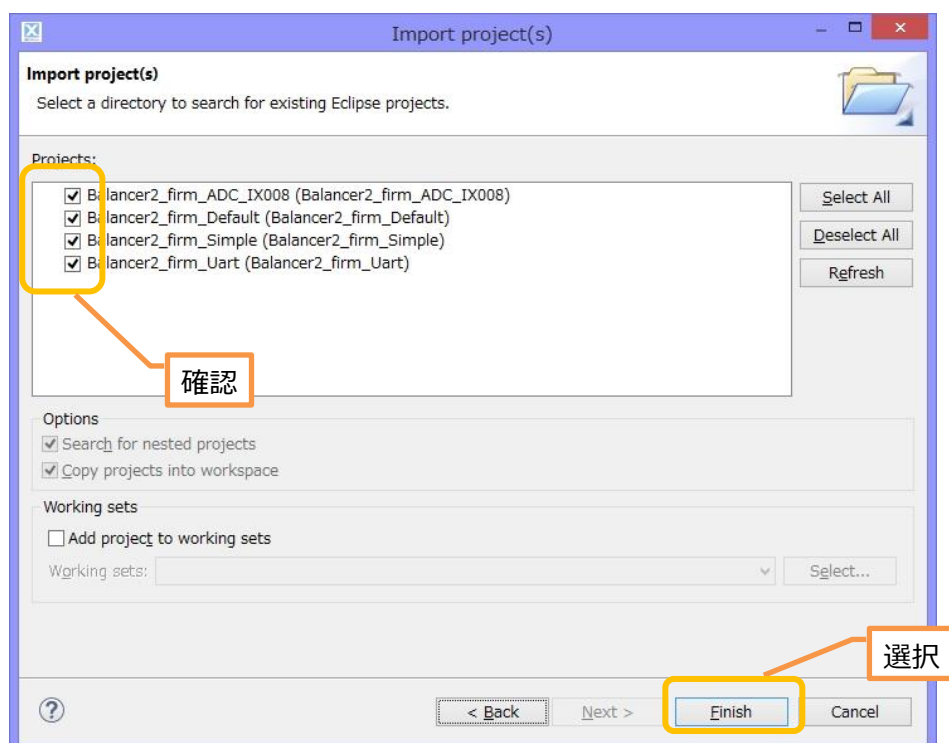
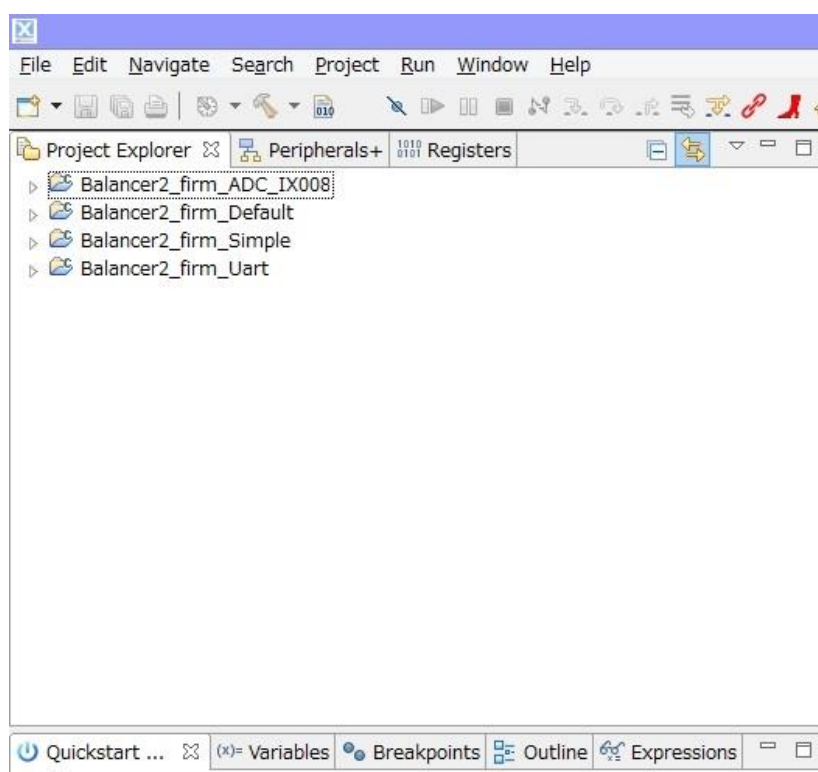
以上で認証は完了です。

7.5. サンプルソースのダウンロードとインポート

サンプルソースとライブラリは、Github にて公開しています。

1) 以下の URL にアクセスします。

https://github.com/vstoneofficial/Balancer2

2) 右下の Download ZIP を選択し、サンプルをダウンロードします。

3) LPCXpresso を開き、Quickstart パネルの>[Start here]>[Import project(s)]を選択します。

4)　Project archive (zip)の Browse…を選択し、ダウンロードした Zip ファイルを選択します。

5)　以下の 4 つの項目にチェックが付いていることを確認し、Finish を選択します。

6) Project Exploer に 4 つのプロジェクトが追加されます。

8. サンプルソースのビルドと実行

ここでは、プロジェクトにインポートしたサンプルソースのビルド～書込・動作確認を行います。

8.1. サンプルソースについて

サンプルソースには以下の 4 種類が含まれます。

A) Balancer2_firm_Default

出荷時のファームウェアです。

VS-C3 での操縦、ログの取得などが含まれます。

B) Balancer2_firm_Simple

制御ループのみのシンプルなファームウェアです。

C) Balancer2_firm_Uart

UART ポートから状態を送信するサンプルです。

Bluetooth モジュール（VS-BT001）などを接続し、PC でログを取得したい場合にご参考ください。

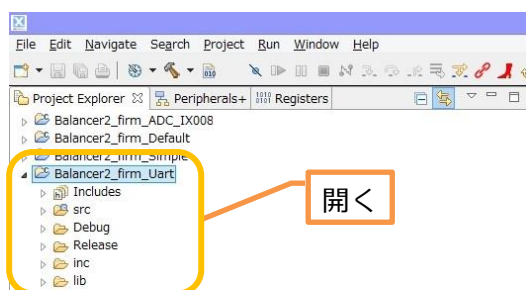
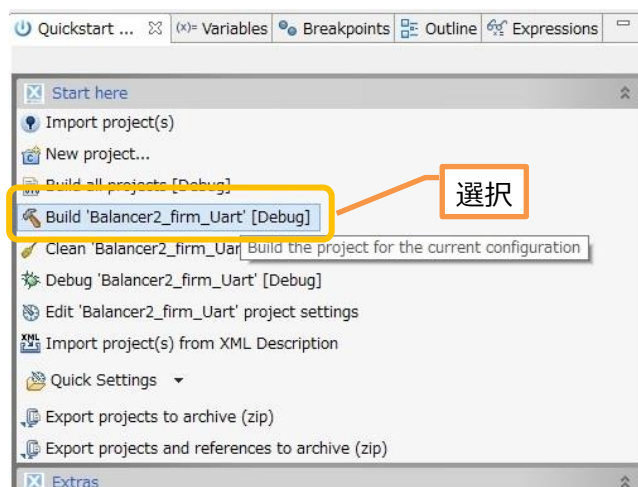
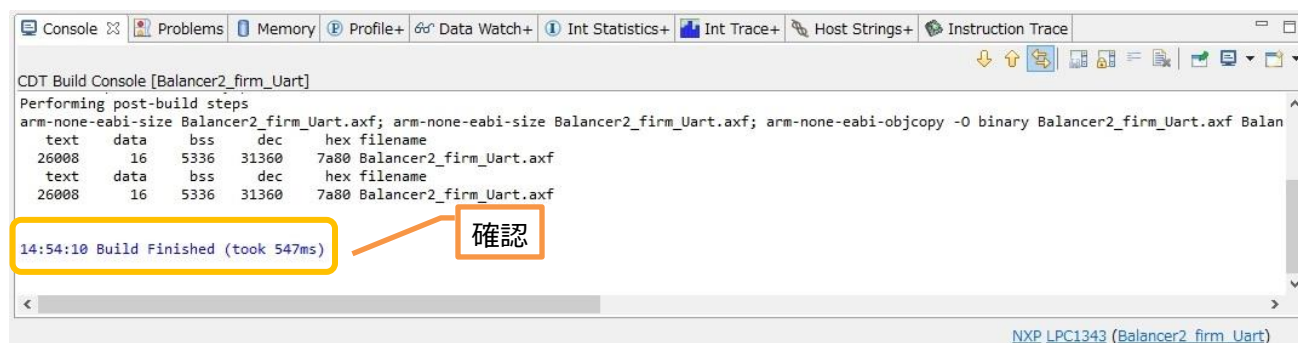
D) Balancer2_firm_ADC_IX008

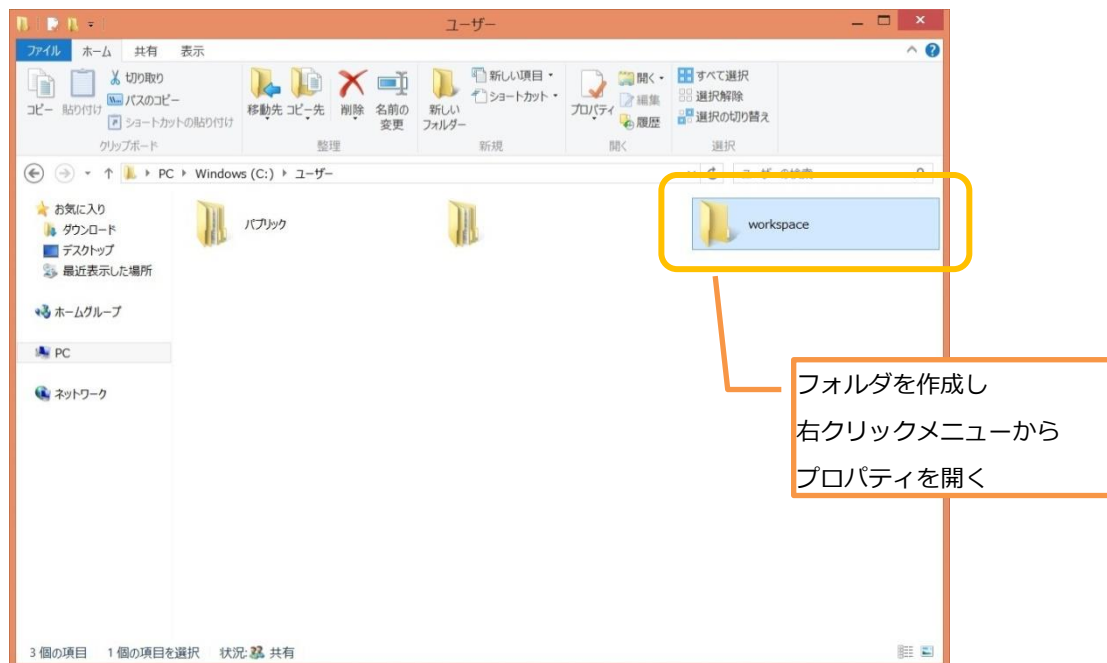
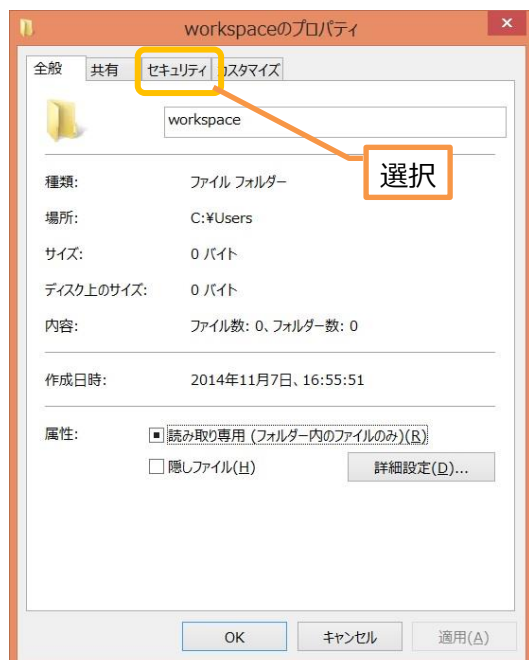
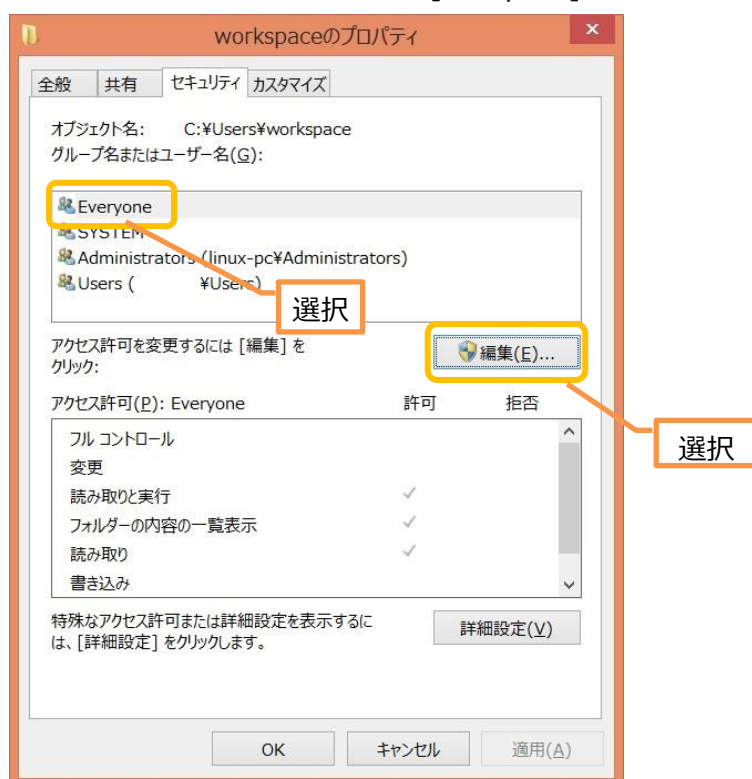
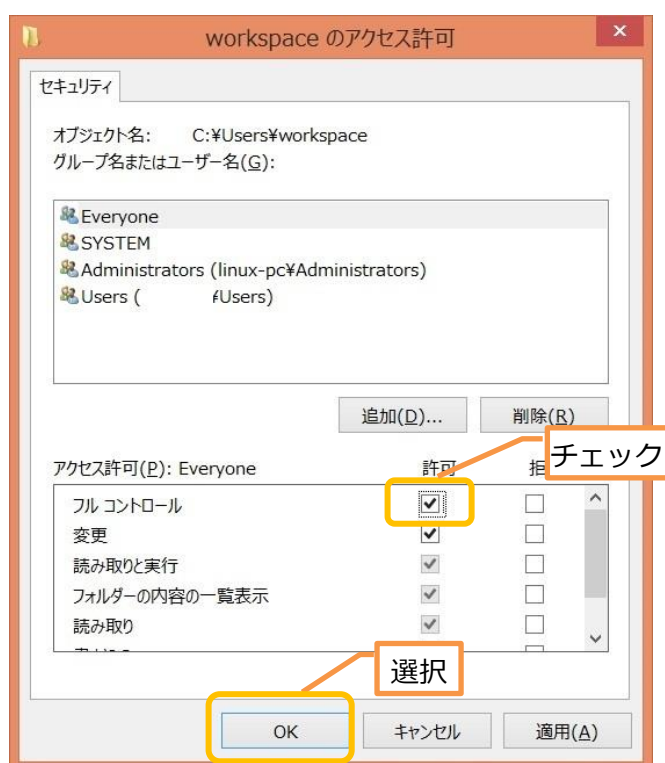
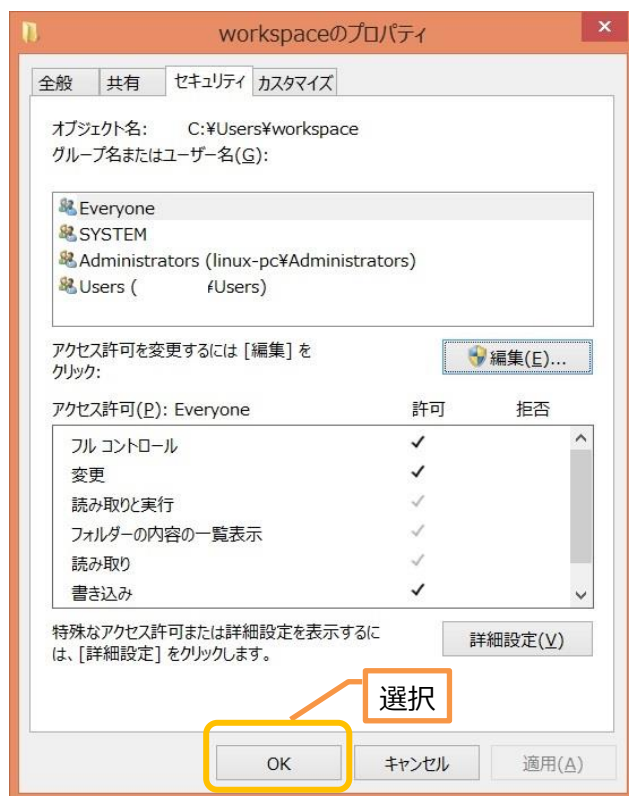
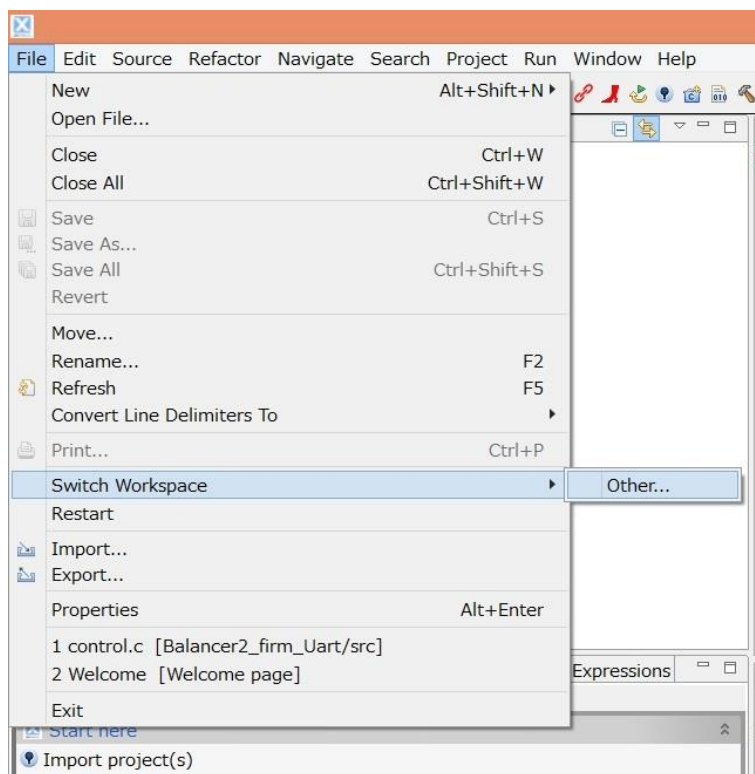
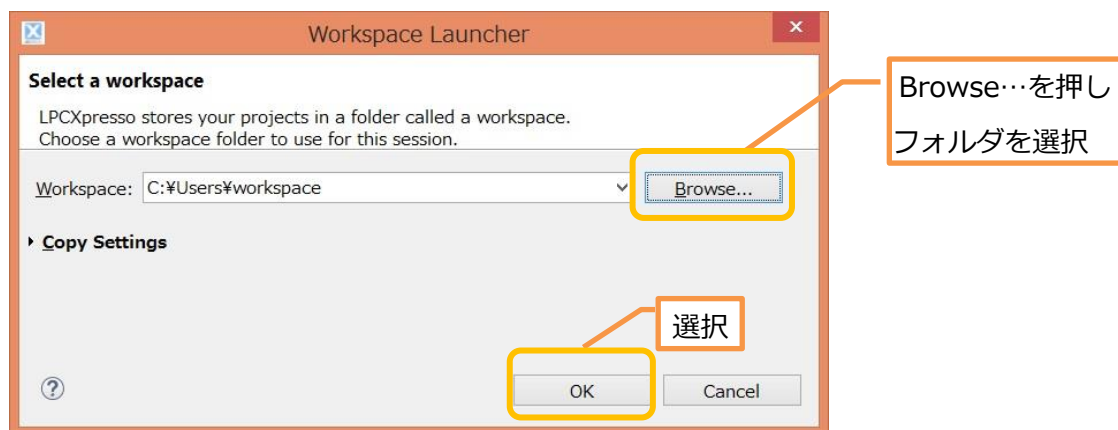
IX ポートにアナログ入力拡張ボード（VS-IX008）を接続し、センサなどの値を取得すサンプルです。

センサなどを追加し拡張する場合にご参考ください。

8.2. サンプルソースのビルド

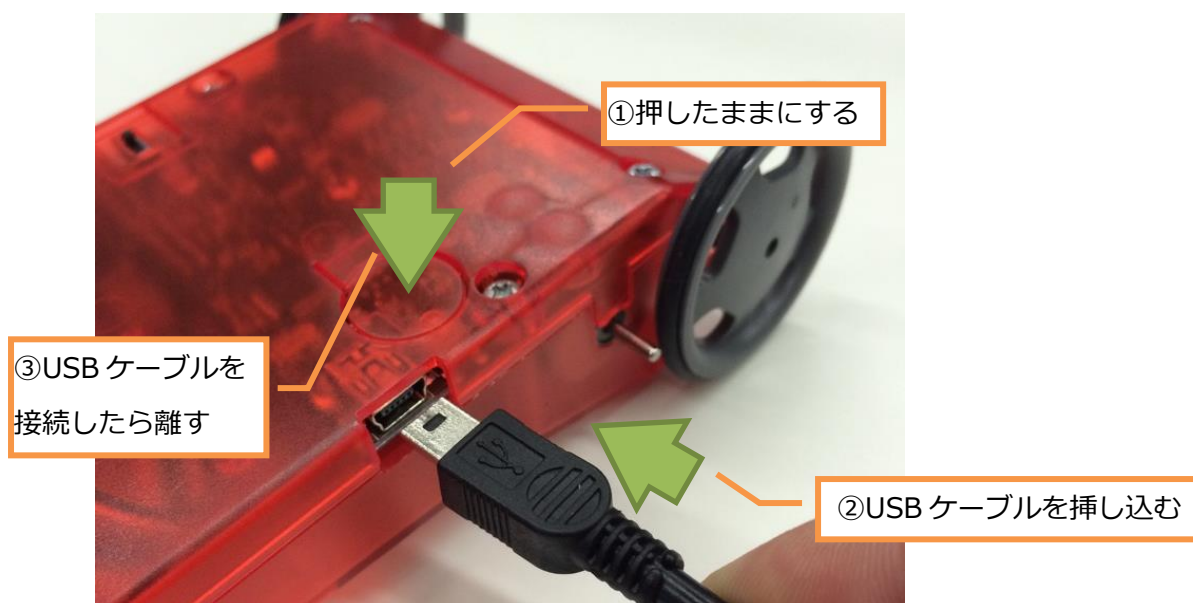
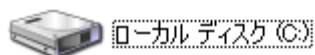
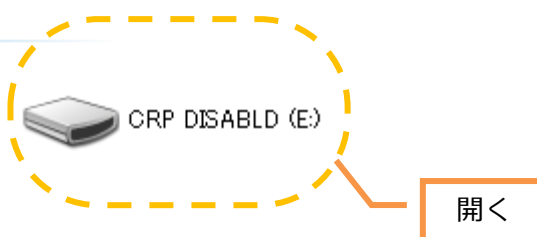
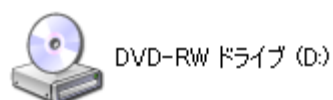
ダウンロードしたサンプルソースをビルドし、本体に書き込める状態にし、USB 経由で本体に書き込みます。ここでは、[Balancer2_firm_Uart]を例に操作方法を説明します。

1) ビルドしたいサンプルを開きます。

2) Quickstart パネルの>[Start here]>[Build Balancer2_firm_Uart [Debug]]を選択すると、ビルドが始まります。([Balancer2_firm_Uart]はプロジェクト名なので、プロジェクトにより名称が変わります)

3) ビルドが完了すると、Console に Build Finised と表示されます。

※ワークスペースまでのファイルパスに日本語が含まれる場合、ビルドエラーになる場合があります。
その場合、以下の手順でワークスペースフォルダの場所を変更してください。

1) C:¥Users¥などに workspace フォルダを作成します。

2) フォルダを右クリックし、セキュリティタブを開きます。

3) グループ名またはユーザー名にて[Everyone]を選択し、編集を開きます。

4) フルコントロールをチェックし、OK を選択します。

5) OK を選択します。

6) LPCXpresso を開き、[File]>[Switch Workspace]>[Other…]を選択します。

7) Browse を押し、先ほど作成したフォルダを選択します。

8) 3.5.(3)の手順で、再度サンプルソースをインポートします。

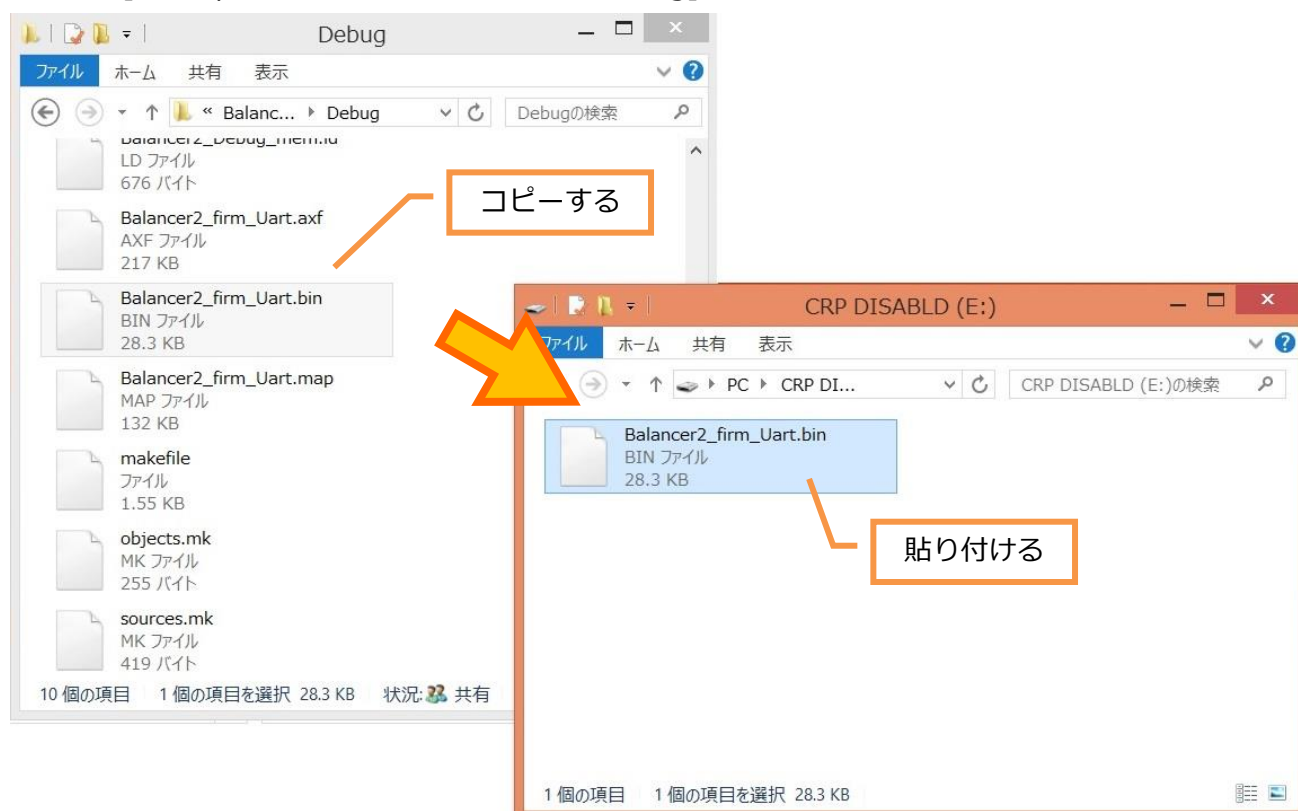
8.3. サンプルソースの書込

ビルドしたサンプルソースを USB 経由で本体に書き込みます。

1) USB ケーブルを外し、本体の**電源を OFF** にします。
2) **操作ボタンを押したまま**、PC と USB ケーブルで接続します。
3) USB ケーブルを接続したら操作ボタンから手を離す。
4) 20 秒程度待つと、PC が本製品を USB メモリなどと同様に「マスストレージデバイス」として認識します。（初めて本製品を PC に接続した場合、認識にしばらく時間がかかります。）

5) CPU ボードが PC に認識されると、「CRP DISABLE」という名称のドライブ名で表示されます。CPU ボードを接続したら「マイコンピュータ」を開いて、この名称のドライブが表示されることを確認し、ダブルクリックしてドライブを開きます。

6)　「CRP DISABLE」を開くと、中に「firmware.bin」というファイルが一つ入っていますので、このファイルを削除します。

7)　プロジェクトフォルダ内の Debug フォルダに作成された、[Balancer2_firm_Uart.bin]をコピーして、「CRP DISABLE」内に貼り付けます。

　今回の場合フォルダは、Debug フォルダは以下の場所に

　[workspace¥Balancer2_firm_Uart¥Debug]になります。

8)　本体から USB ケーブルを抜き、再度電源を入れると、書き込んだプログラムに従って本体が動作します。

9. バランサー2 SDK の仕様

ここでは、バランサー2 を制御するためのライブラリ「バランサー2 SDK」の仕様について、サンプルソースに沿って説明します。

9.1. 概要説明

「バランサー2 SDK」は、ロボットの CPU ボードを PC から通信制御するライブラリです。対応開発環境は、Microsoft Visual C++で、CPU ボードの制御は本ライブラリの関数からメモリマップを読み書きし、「モータの駆動」「センサ情報の読み込み」などを行います。

9.2. 動作環境

本ライブラリの動作環境は以下の通りです。

・OS：Windows XP SP3 以降/Vista/7/8

・対応言語：Microsoft Visual C++ 6.0 以降

・インターフェース：USB

9.3. サンプルソースについて

サンプルソースには以下の 4 種類が含まれます。

A) Balancer2_SDK_Sample.cpp

SDK の各機能を簡易的に使用したソースコードです。

B) BB022HIDSDK.h

「バランサー2 SDK」のヘッダーファイルです。SDK を使用するときに必要となります。

C) BB022HIDSDK.dll

「バランサー2 SDK」の DLL ファイルです。SDK を使用するときに必要となります。

D) BB022HIDSDK.lib

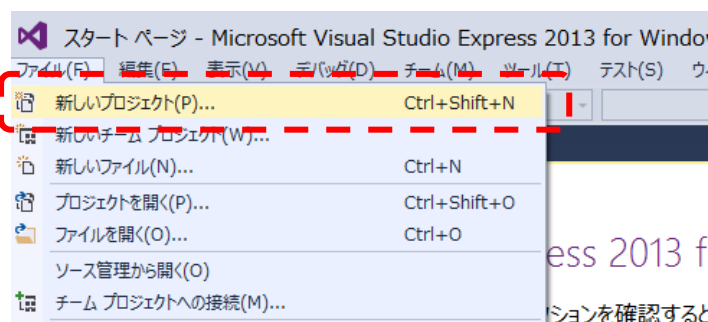
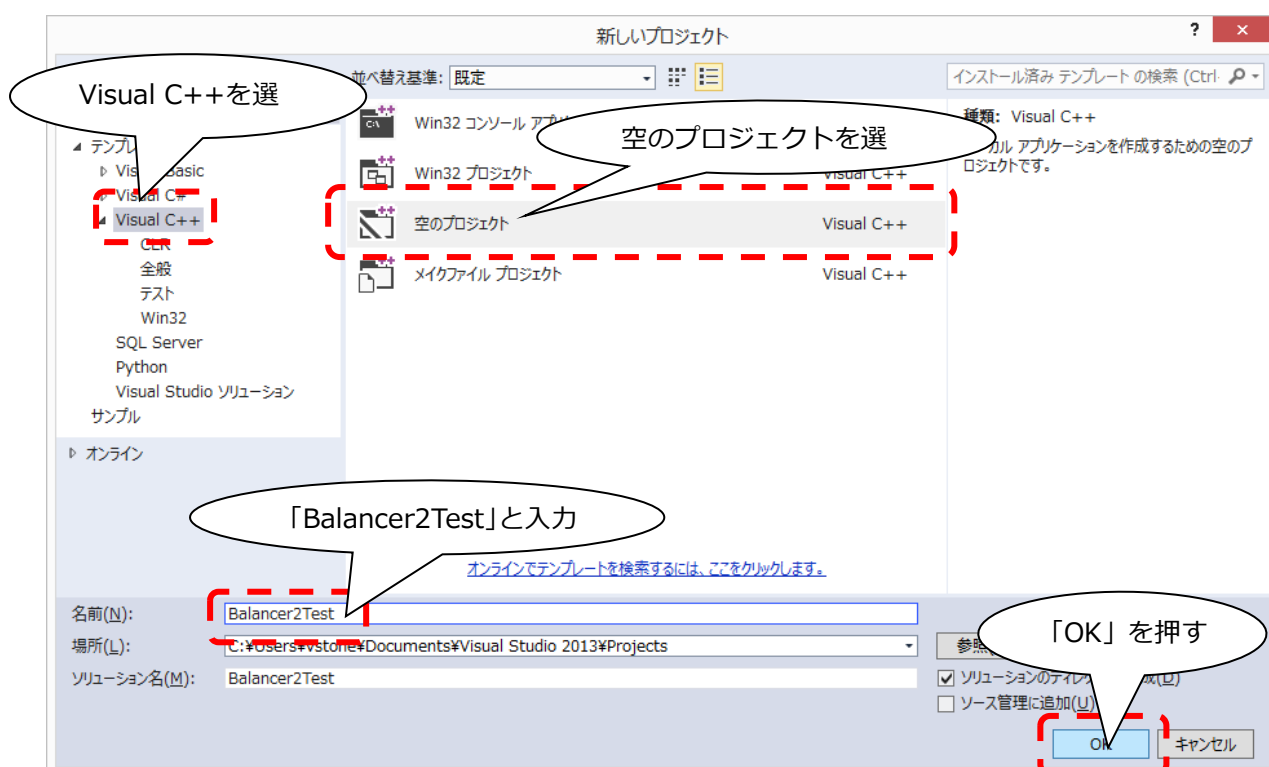
「バランサー2 SDK」のライブラリです。SDK を使用するときに必要となります。

9.4. サンプルソースの実行方法

ここでは、Microsoft Visual Studio Express 2013 for Windows Desktop を使用した実行方法を説明します。開発環境については以下のサイトよりダウンロードし、インストールを行ってください。

http://www.visualstudio.com/ja-jp/products/visual-studio-express-vs.aspx

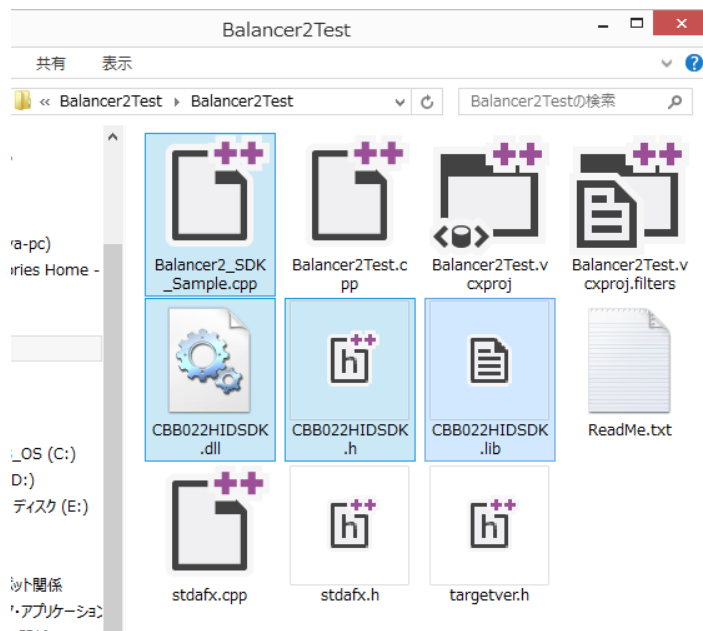
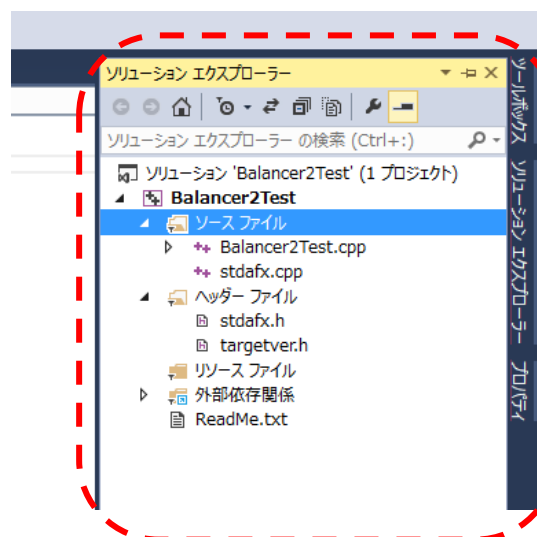
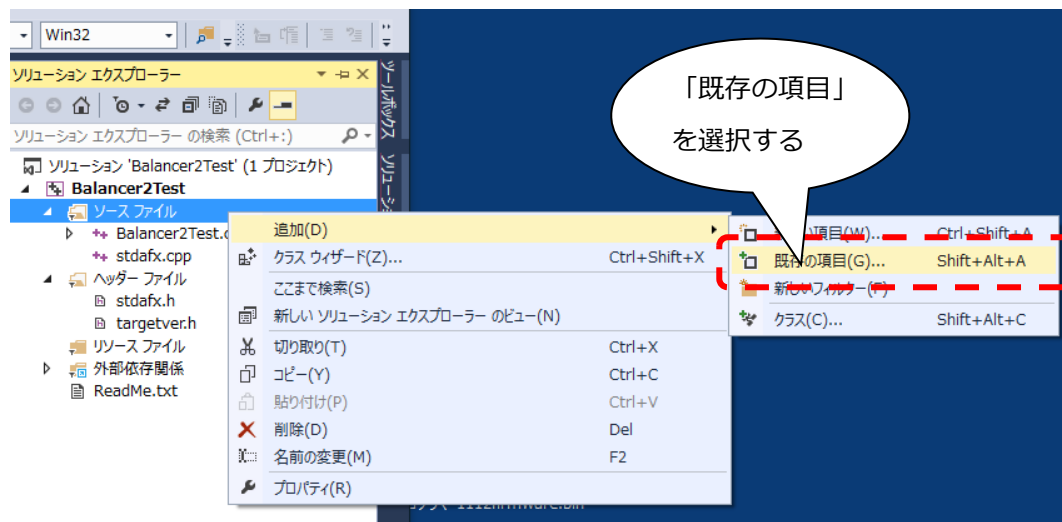
A) プロジェクトの作成

まず、VisualStudio を開き、新しいプロジェクトを作成します。

次に、左枠の「テンプレート」>「VisualC++」を選択し、中央枠の「空のプロジェクト」を選択します。また、プロジェクト名は「Balancer2Test」と入力し、「OK」を押します。

B)　ファイルの追加

サンプルソース等のファイルをプロジェクトに追加します。内容については 8.3 章をご覧ください。

ダウンロードしファイルをプロジェクトにコピーします。コピー先は以下の通りです。

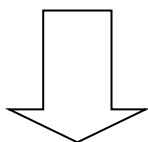
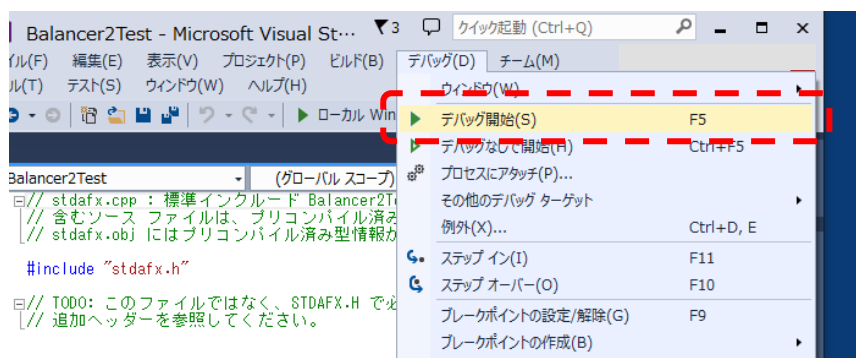
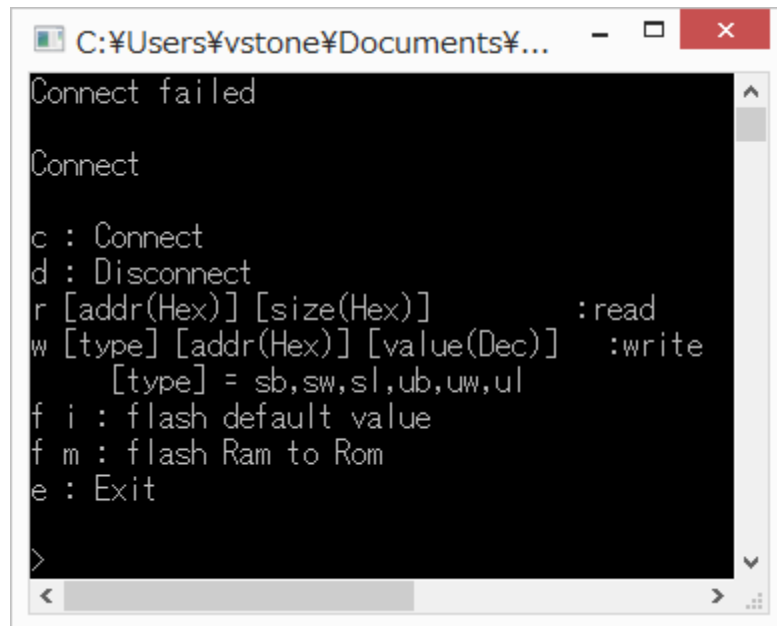
「C:¥Users¥ユーザ名¥Documents¥Visual Studio 2013¥Projects
¥Balancer2Test¥Balancer2Test」

また、コピーするファイルは 8.3 項で紹介した 4 つのファイルです。

ソリューションエクスプローラーの画面を表示させます。表示の仕方は、「ツール>ソリューションエクスプローラー」を押すとできます。既に表示されている場合はそのまま表示させます。

ソリューションエクスプローラーの「ソースファイル」を右クリックし、「追加>既存の項目」をクリックします。

「Balancer2_SDK_Sample.cpp」を追加します。

この手順と同様に、ソリューションエクスプローラーの「ヘッダーファイル」を右クリックし、「追加>既存の項目」で「CBB022HIDSDK.h」を追加します。

C) 実行ファイルの起動

アプリケーションソフトをビルドと同時に起動します。

「デバッグ>デバッグ開始」をクリックするとコンソール画面が現れます。

D) サンプルプログラムの概要

コンソール画面を開いた状態で、バランサー2 との接続待ちとなり、未接続の場合「Connect failed」と表示されます。接続した場合「Connect」と表示され、その下にコマンドの説明が表示されます。

コマンドの入力方法は以下の通りです。

- 「c」
  CPU ボードとの通信を開始する
- 「d」
  CPU ボードとの通信を切断する
- 「r [アドレス(16 進数)] [サイズ(16 進数)] 」
  メモリマップのアドレスからサイズ分のデータを読み込む
- 「w [型※] [アドレス(16 進数)] [値(10 進数)]」
  メモリマップのアドレスから型に合わせた値を RAM に書き込む

※型の指定方法は、「sb(byte)」「sw(word)」「sl(long」「ub(ubyte)」「uw(uword)」「ul(ulong)」のいずれかの各 2 文字を入力します。

- 「f　i」

  Flash ROM を初期状態にします。

- 「f　m」

  RAM の内容を Flash ROM に書き込みます。Flash ROM に書き込むことで、本体の電源を切っても書き込んだ内容は消えません。

- 「e」

  アプリケーションソフトを終了します。

9.5. 関数の仕様説明

ライブラリに備わっている関数の概要は以下の通りです。

・**int CWRC_Connect(USHORT pid)**

　CPU ボードと通信を開始します。戻り値は、正しく通信を開始できれば 1、CPU ボードが見つからない、または他のプログラムと既に通信しているなどにより通信を開始できなければ 0 を返します。引数は、プロダクト ID を与えます。

・**void CWRC_Disconnect()**

　CPU ボードとの通信を終了します。

・**int CWRC_ReadMemMap(int staddr, int sz)**

　CPU ボードから読み込むメモリマップのアドレスを指定します。実際に読み込みを行なうのは ReadExecute 関数を実行したときです。引数は、'int staddr'に読み込みを開始するメモリマップアドレス、'int sz'に読み込むメモリサイズ（byte 単位）を与えます。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

　メモリマップのアドレス概要は、10.7 項「ライブラリメモリマップ」を参照してください。

・**int CWRC_ReadExecute()**

　ReadMemMap 関数で指定したアドレスを CPU ボードから読み込み、ライブラリ側のメモリマップ情報を更新します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int CWRC_WriteMemMap(int staddr,BYTE data)

CPU ボードのメモリマップに書き込むデータをセットします。実際に書き込みを行なうのは WriteExecute 関数を実行したときです。引数は、'int staddr'にはデータを書き込むアドレス、'BYTE data'には、書き込む数値（byte 単位）を、それぞれ与えます。

戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

2byte の連続したメモリブロックに数値を書き込む場合は、上位バイト、下位バイトごとに分けて 2 回関数を実行してください。また CPU ボードのバイトオーダーはビッグエンディアンのため、番号の小さい側のアドレスから上位バイトを割り当ててください。

・int CWRC_WriteExecute()

WriteMemMap 関数で指定したアドレスを CPU ボードから読み込み、ライブラリ側のメモリマップ情報を更新します。戻り値は、失敗した場合は「0」、成功した場合は「1」を返します。

・int CWRC_FlashExecute()

WriteExecute 関数で更新したメモリマップ情報を RAM から FlashROM へと転送します。戻り値は、失敗した場合は「0」、成功した場合は「1」を返します。

・int CWRC_InitializeExecute()

メモリマップの初期情報を FlashROM に書き込みます。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・char GetMem_SByte(UWORD adr)

メモリマップから符号付 char（1byte）の情報を取得します。引数は、’UWORD adr’には読み込むアドレスを与えます。戻り値は、読み込んだメモリマップの数値です。

・short GetMem_SWord(UWORD adr)

メモリマップから符号付 short（2byte）の情報を取得します。引数は、’UWORD adr’には読み込むアドレスを与えます。アドレスは 2byte のうち上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・long GetMem_SLong(UWORD adr)

　メモリマップから符号付 long（4byte）の情報を取得します。引数は、' UWORD adr'には読み込むアドレスを与えます。アドレスは 4byte のうち最上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・long long GetMem_SLongLong(UWORD adr)

　メモリマップから符号付 long long（8byte）の情報を取得します。引数は、' UWORD adr'には読み込むアドレスを与えます。アドレスは 8byte のうち最上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・BYTE GetMem_UByte(UWORD adr)

　メモリマップから符号無し char（1byte）の情報を取得します。引数は、' UWORD adr'には読み込むアドレスを与えます。戻り値は、読み込んだメモリマップの数値です。

・unsigned short GetMem_UWord(UWORD adr)

　メモリマップから符号無し short（2byte）の情報を取得します。引数は、' UWORD adr'には読み込むアドレスを与えます。アドレスは 2byte のうち上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・unsigned long GetMem_ULong(UWORD adr)

　メモリマップから符号無し long（4byte）の情報を取得します。引数は、' UWORD adr'には読み込むアドレスを与えます。アドレスは 4byte のうち最上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・unsigned long long GetMem_ULongLong(UWORD adr)

　メモリマップから符号無し long long（8byte）の情報を取得します。引数は、' UWORD adr'には読み込むアドレスを与えます。アドレスは 8byte のうち最上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・float GetMem_Float(UWORD adr)

メモリマップから浮動小数点型 float（4byte）の情報を取得します。引数は、'UWORD adr'には読み込むアドレスを与えます。アドレスは 4byte のうち最上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・double GetMem_Double(UWORD adr)

メモリマップから浮動小数点型 double（8byte）の情報を取得します。引数は、'UWORD adr'には読み込むアドレスを与えます。アドレスは 8byte のうち最上位 byte のアドレスを指定します。戻り値は、読み込んだメモリマップの数値です。

・int SetMem_SByte(UWORD adr, char data)

メモリマップの指定のアドレスに、符号付 char（1byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'char data'には書き込むデータの実数を与えます。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_SWord(UWORD adr, short data)

メモリマップの指定のアドレスに、符号付 short（2byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'short data'には書き込むデータの実数を与えます。アドレスは 2byte のうち上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_SLong(UWORD adr, long data)

メモリマップの指定のアドレスに、符号付 long（4byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'long data'には書き込むデータの実数を与えます。アドレスは 4byte のうち最上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_SLongLong(UWORD adr, long long data)

メモリマップの指定のアドレスに、符号付 long long（8byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'long long data'には書き込むデータの実数を与えます。アドレスは 8byte のうち最上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_UByte(UWORD adr, BYTE data)

メモリマップの指定のアドレスに、符号無し char（1byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'BYTE data'には書き込むデータの実数を与えます。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_UWord(UWORD adr, unsigned short data)

メモリマップの指定のアドレスに、符号無し short（2byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'unsigned short data'には書き込むデータの実数を与えます。アドレスは 2byte のうち上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_ULong(UWORD adr, unsigned long data)

メモリマップの指定のアドレスに、符号無し long（4byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'unsigned long data'には書き込むデータの実数を与えます。アドレスは 4byte のうち最上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_ULongLong(UWORD adr, unsigned long long data)

メモリマップの指定のアドレスに、符号無し long long（8byte）の情報を書き込みます。引数は、' UWORD adr'には読み込むアドレスを、'unsigned long long data'には書き込むデータの実数を与えます。アドレスは 8byte のうち最上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_Float(UWORD adr, float data)

メモリマップの指定のアドレスに、浮動小数点型 float（4byte）の情報を書き込みます。引数は、’UWORD adr’には読み込むアドレスを、’float data’には書き込むデータの実数を与えます。アドレスは 8byte のうち最上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

・int SetMem_Double(UWORD adr, double data)

メモリマップの指定のアドレスに、浮動小数点型 double（8byte）の情報を書き込みます。引数は、’UWORD adr’には読み込むアドレスを、’double data’には書き込むデータの実数を与えます。アドレスは 8byte のうち最上位 byte のアドレスを指定します。戻り値は、Connect 関数を実行していない、またはメモリマップの範囲外のアドレスを指定した場合は 0、それ以外の場合は 1 を返します。

9.6. メモリマップの概要

CPU ボードのメモリマップは 10.7 章「ライブラリメモリマップ」をご参照ください。メモリマップは、アドレスによってサイズや符号が混在しており、また、内容を書き換えることで CPU ボードの動作が変化する場合があります。もし CPU ボードの状態を元に戻せなくなったら、7 章を参考にファームウェアを書き直してください。

10.　トラブルシューティング

10.1. 本体について

Q1. 電源が入らない

A1. 電池の向きに間違いがないか確認して下さい。

A2. 電池は新しいもの（充電池の場合、満充電のもの）をお使いですか？

A3. 本製品は単 3 アルカリ乾電池または単 3 ニッケル水素電池のみに対応しております。マンガン電池には対応しておりません。

Q2. 倒立できない、または倒立が不安定

A1. タイヤの遊びが多い場合や、遊びが全くない場合は、正しく制御することができません。

A2. タイヤの前後方向の遊びが 0.5mm 以下になるようにしてください。（組立説明書の手順 10 番をご確認ください。）

Q3. モータ軸が回らない

A1. 転倒時、まれに電池が外れてしまうことがあります。電池が正しく入っているかご確認ください。

A2. 一度電池を取り外し、再度入れ直してみてください。

A3. モータの端子部が基板に接しているか確認して下さい。

A4. 組み立て時にモータの端子を起こさないと、基板との接触不良によりモータが回転しない場合があります。組立説明書の手順 04 番からやり直してください。

10.2. バランサー2 プログラマについて

「バランサー2 プログラマ」が正しく動作しない場合は、以下の項目をご確認ください。それでも解決しない場合は、ドキュメント末尾の「製品に関するお問い合わせ」の連絡先にお問い合わせください。

Q4. USB ケーブルを挿しているが、操作画面の左上が「未接続」の表示のまま変わらない

A1. USB ケーブルを挿し直してお試しください。それでも改善しない場合は、USB ケーブルを変えてお試し下さい。

Q5.「操作モード」で矢印キーを押しているが、バランサー2 本体が動かない

A1. 2.5 をご確認の上､｢操作モード｣のチェックボックスにチェックが入っているかをご確認ください。

Q6. バランサー2 を動作させても、すぐに倒れてしまう

A1. 電池を新品にしてお試し下さい。それでも倒れてしまう場合は、ゲインを初期値にしてお試し下さい。

※インストール先（C:¥Program Files (x86)¥Balancer2Programmer）の中に、「init_gain.b2g」があるので、4.16 項「ゲインの読込」をご確認の上、「ファイルから読込」ボタンを押し、「init_gain.b2g」を読み込んで、「PC->RAM」を押し、その後「RAM->ROM」ボタンを押すことで初期値に戻ります。

Q7. 線がグラフに表示されない

A1. 画面左上が「接続」になっているか、また、2.3 をご確認の上、グラフを表示させてください。

Q8. ログのデータが保存できない、または、正常な値が書き込まれない

A1. RAM 内容量の都合上、「ログカウンタ」までしかログが保存できません。ご了承ください。

Q9. リストを右クリックして、値を書き換えることができない

A1. 型（char 等）に合う数値かご確認ください。

A2. 整数型(short 等)に実数型(0.1 等)を入力していないかご確認ください。

A3. 全角文字が含まれていないかご確認ください。

Q10.　バランサー２が倒立しなくなった・動作しなくなった

A1. 型（char 等）に合う数値かご確認ください。

A2. 整数型(short 等)に実数型(0.1 等)を入力していないかご確認ください。

A3. 全角文字が含まれていないかご確認ください。

11.　参考

11.1. ジャイロセンサ仕様

A)　型番　　　　　　　村田製作所製　ENC-03RC/D

B)　検出範囲　　　　　±1.667π[rad/sec]（±300[deg/sec]）

C)　分解能　　　　　　0.006680465 [rad/sec]（1.202483675[deg/sec]）

D)　取得周期　　　　　2[msec]

11.2. ロータリーエンコーダ仕様

A)　車輪減速比　　　　21:1（モータ軸径φ2[mm]、タイヤ径φ42mm）

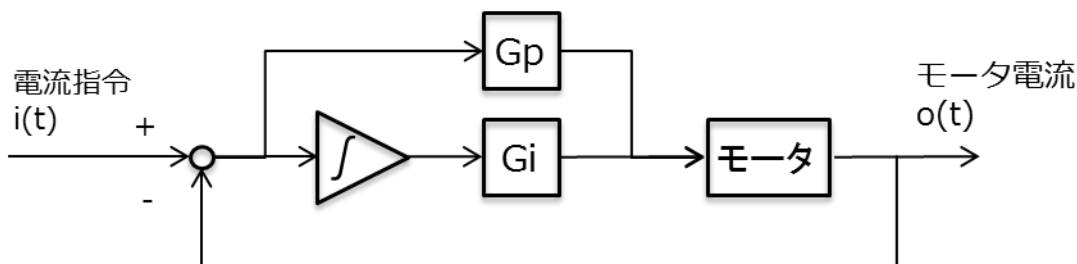
B)　分解能　　　　　　0.01246664[rad/sec]

C)　最大検出速度　　　100[kpps]

11.3. 電流制御仕様

A)　電流検出分解能　　1.61133[mA]

B)　制御周期　　　　　1[msec]

C)　ブロック線図

i(t)　　電流指令値（メモリマップ 8,10 番地）

o(t)　　モータ電流測定値（メモリマップ 4,6 番地）

Gp　　電流制御 P ゲイン（メモリマップ 288 番地）

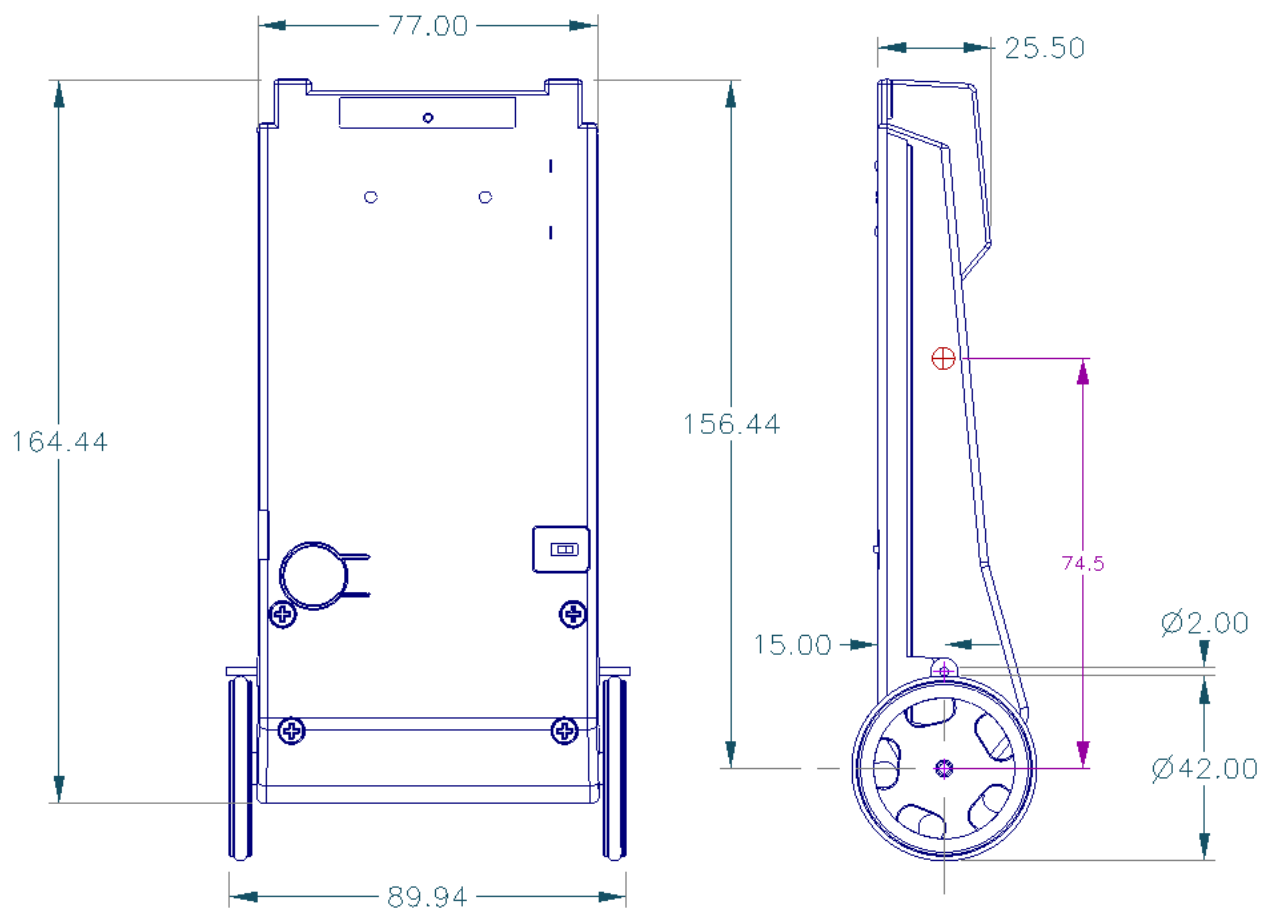
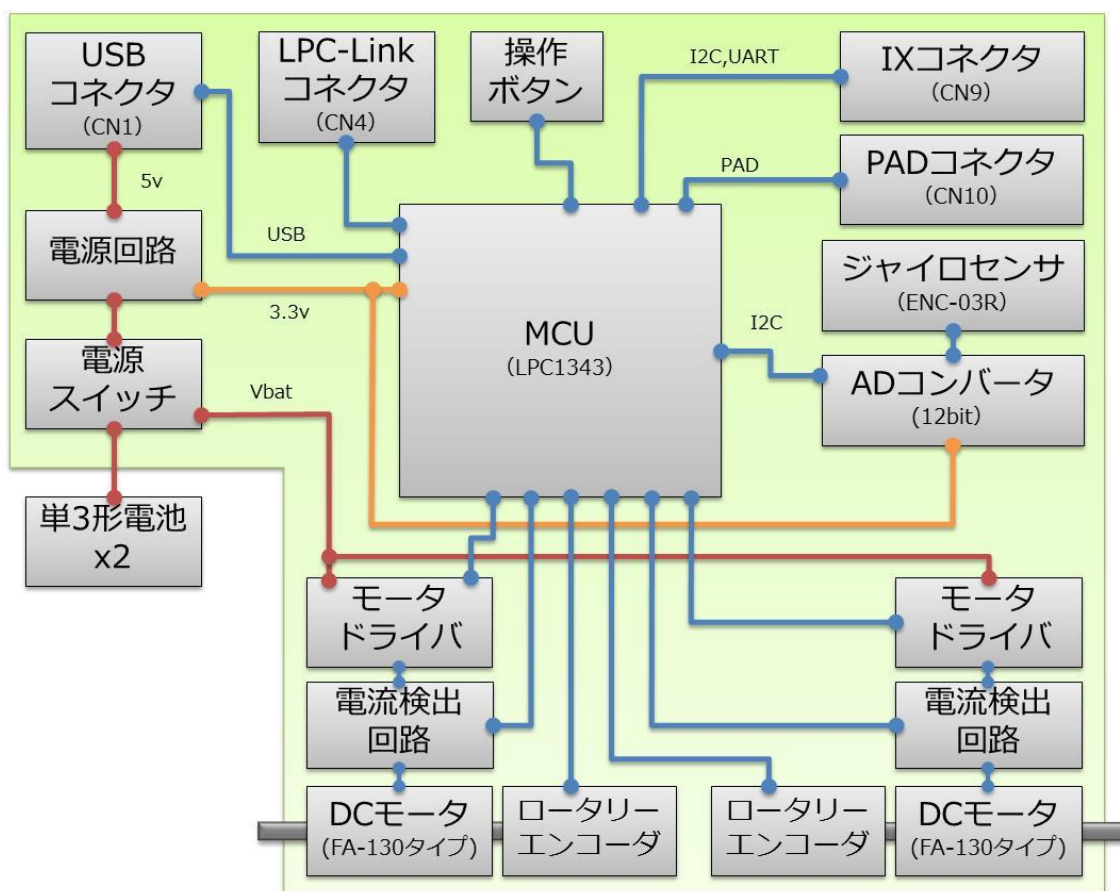
Gi　　電流制御 I ゲイン（メモリマップ 296 番地）

11.4. 寸法図

重心位置

車輪中心軸から X : 0[mm]　Y : 74.5[mm]

11.5. ブロック図

USB
コネクタ
(CN1)
LPC-Link
コネクタ
(CN4)
操作
ボタン
I2C,UART
IXコネクタ
(CN9)
PAD
PADコネクタ
(CN10)
5v
電源回路
USB
3.3v
ジャイロセンサ
(ENC-03R)
MCU
(LPC1343)
I2C
電源
スイッチ
Vbat
ADコンバータ
(12bit)
単3形電池
x2
モータ
ドライバ
モータ
ドライバ
電流検出
回路
電流検出
回路
DCモータ
(FA-130タイプ)
ロータリー
エンコーダ
ロータリー
エンコーダ
DCモータ
(FA-130タイプ)

11.6. 拡張コネクタ

A) CN9（IX コネクタ）

I2C・UART 機器を拡張する際に使用できるコネクタです。

弊社製の拡張ボードを使用する場合、別売のフ「10 芯コネクタ+フラットケーブル（圧着済み）」をご使用ください。

1:TXD(UART)	6:SDA(I2C)
2:RXD(UART)	7:Vbat
3:NC	8:+５v
4:NC	9:+3.3v
5:SCL(I2C)	10:GND

B) CN10 （PAD コネクタ）

VS-C3 専用ポートになります。ユーザプログラムからは使用できません。

C) CN4（LPC-Link）

NXP 社製デバッガ　LPC-Link を接続するためのコネクタです。

コネクタは実装されておりませんので、JST 社製 S8B-ZR-SM3A を実装いただくか、ケーブルをはんだづけしてお使いください。

1:3.3v	5:NC
2:SWDIO	6:RESET
3:SWCLK	7:NC
4:SWO	8:GND

11.7. ライブラリメモリマップ

アドレス 10進数	名称		型	サイズ (byte)	R/W	初期値	備考
0	プロダクトID	Product_ID	unsigned short	2	R	0x2080	
2	ファームウェア バージョン	Version	unsigned char	1	R	1	
3	モード	MODE	unsigned char	1	R	0	0:モータ電源OFF 1:モータ電源ON
4	モータ電流測定値	M_CURRENT_L	short	2	R	0	mA
6		M_CURRENT_R	short	2	R	0	
8	モータ電流指令値値	T_CURRENT_L	short	2	R	0	mA
10		T_CURRENT_R	short	2	R	0	
12	電流オフセット	CURRENT_OFFSET_L	short	2	R	0	mA
14		CURRENT_OFFSET_R	short	2	R	0	
16	本体角速度	BODY_ANGULAR_SPD	double	8	R	0.0	rad/sec
24	本体角度	BODY_ANGLE	double	8	R	0.0	rad
32	本体角速度オフセット	BODY_ANGULAR_SPD_OFFSET	double	8	R	0.0	rad/sec
40	ホイール角速度　左	WHEEL_ANGULAR_SPD_L	double	8	R	0.0	rad/sec
48	ホイール角速度　右	WHEEL_ANGULAR_SPD_R	double	8	R	0.0	rad/sec
56	ホイール角度　左	WHEEL_ANGLE_L	double	8	R	0.0	rad
64	ホイール角度　右	WHEEL_ANGLE_R	double	8	R	0.0	rad
72	目標速度	T_SPD_L	double	8	R/W	0.0	rad/sec
80		T_SPD_R	double	8	R/W	0.0	
88	エンコーダ	ENC_L	long long	8	R	0	4逓倍の値
96		ENC_R	long long	8	R	0	
104	ジャイロ	GYRO_DATA	short	2	R	0	ジャイロセンサ　ADCデータ
106	ADC(電流)	ADC_C_LA	unsigned short	2	R	0	モータ電流値　ADCデータ
108		ADC_C_LB	unsigned short	2	R	0	
110		ADC_C_RA	unsigned short	2	R	0	
112		ADC_C_RB	unsigned short	2	R	0	
114	VS-C3	PAD_BTN	unsigned short	2	R	0	MSB　左 | 下 | 右 | 上 |ｽﾀｰﾄ| R3 | L3 |ｾﾚｸﾄ LSB □ | × | ○ | △ | R1 | L1 | R2 | L2
116		PAD_AN_RX	char	1	R	0	-127~128
117		PAD_AN_RY	char	1	R	0	-127~128
118		PAD_AN_LX	char	1	R	0	-127~128
119		PAD_AN_LY	char	1	R	0	-127~128
120	ユーザエリア	USER_ARIA1	double	8	R/W	0.0	ファームウェアから任意の数値を読み書き可能
128		USER_ARIA2	double	8	R/W	0.0	ログの取得やUSB経由で数値をモニタリングしたいときに使用する
136		USER_ARIA3	double	8	R/W	0.0	
144		USER_ARIA4	double	8	R/W	0.0	
152		USER_ARIA5	double	8	R/W	0.0	
160		USER_ARIA6	double	8	R/W	0.0	
168		USER_ARIA7	double	8	R/W	0.0	
176		USER_ARIA8	double	8	R/W	0.0	

アドレス 10進数	名称		型	サイズ (byte)	R/W	初期値	備考
184	キャリブレーション カウンタ	CALIBRATISON_CNT	unsigned short	2	R/W	0	キャリブレーション用カウンタ
186	キャリブレーション 時間	CALIBRATISON_TIME	unsigned char	1	R/W	5	キャリブレーションを行う時間
187	ログMODE	LOG_MODE	unsigned char	1	R/W	0	B0: RECODIONG /開始時にセット、終了時に自動的にクリア B1: RESET /ログ用メモリをクリア B7: ENABLE /ログ取得を有効化 ログ用メモリはROM書換用メモリと同一なので、ROM書換を行うと消去される B0をセットしてROMに書いておけば、動作開始直後からログを取得できる
188	ログDivider	LOG_DIV	unsigned short	2	R/W	1	分周値　500/(LOG_DIV)Hzで取得 0:無効 1:500Hz 10:50Hz
190	ログカウンタ	LOG_CNT	unsigned short	2	R/W	0	ログカウンタ(byte数)
192	ログフラグ	LOG_FLG_1	unsigned long long	8	R/W	0	メモリマップのアドレス0番地(0bit)～63番地(63bit)を保存するか設定、1:有効、0:無効
200		LOG_FLG_2	unsigned long long	8	R/W	0	メモリマップのアドレス64番地(0bit)～127番地(63bit)を保存するか設定、 1:有効、0:無効
208		LOG_FLG_3	unsigned long long	8	R/W	0	メモリマップのアドレス128番地(0bit)～191番地(63bit)を保存するか設定、 1:有効、0:無効
216	倒立制御ゲイン	GAIN_BODY	double	8	R/W	1.0	本体ゲイン
224		GAIN_WHEEL	double	8	R/W	1.0	タイヤゲイン
232		GAIN_WHEEL_CORRELATION	double	8	R/W	1.0	タイヤ相関ゲイン
240		GAIN_BODY_ANGLE	double	8	R/W	-36400.0	本体角度
248		GAIN_BODY_ANGULAR_SPD	double	8	R/W	-5460.0	本体角速度
256		GAIN_WHEEL_ANGLE	double	8	R/W	1320.0	タイヤ角度
264		GAIN_WHEEL_ANGULAR_SPD	double	8	R/W	396.0	タイヤ角速度
272		GAIN_WHEEL_CORRELATION_ANGLE	double	8	R/W	2400.0	左右タイヤ間　角度偏差
280		GAIN_WHEEL_ CORRELATION_ANGULAR_SPD	double	8	R/W	360.0	左右タイヤ間　角速度偏差
288	モータ電流制御ゲイン	GAIN_CURRENT_P	double	8	R/W	0.00023	Pゲイン
296		GAIN_CURRENT_I	double	8	R/W	0.01200	Iゲイン
304	ジャイロHPF係数	GAIN_GYRO_HPF_COF	double	8	R/W	0.99950	ジャイロHPF係数
312	オプショナルゲイン	GAIN_OPTION1	double	8	R/W	0.0	ユーザが独自の倒立制御を実装する際に、任意にゲインをFlashRomに記憶させることができる
320		GAIN_OPTION2	double	8	R/W	0.0	
328		GAIN_OPTION3	double	8	R/W	0.0	
336		GAIN_OPTION4	double	8	R/W	0.0	
344		GAIN_OPTION5	double	8	R/W	0.0	
352		GAIN_OPTION6	double	8	R/W	0.0	

11.8. ライブラリの関数

11.8.1. 共通

- `void Balancer2Init();`
  初期化

  引数　　なし

  戻値　　なし

- `void setLED(unsigned char value);`
  LED 点灯

  引数　　value　　bit0：オレンジ LED　，　bit1：赤 LED

  戻値　　なし

- `unsigned char CycleProc();`
  周期制御処理

  引数　　なし

  戻値　　0：停止中　　1：取得中

- `unsigned char getSW();`
  スイッチ状態取得

  引数　　なし

  戻値　　1：押下

- `void Wait(unsigned int msec);`
  時間待ち

  引数　　msec　　待ち時間を msec 単位で指定

  戻値　　なし

11.8.2. Senser

- `void CurrentControlEnable();`
  電流制御有効

  引数　　なし

  戻値　　なし

- `void CurrentControlDisable();`
  電流制御無効

  引数　　なし

  戻値　　なし

- `void ClearEncVariables();`
  エンコーダ用変数初期化

  引数　　なし

  戻値　　なし

- `void ClearCurrentVariables();`
  電流制御用変数初期化

  引数　　なし

  戻値　　なし

- `void ClearGyroVariables();`
  ジャイロセンサ用変数初期化

  引数　　なし

  戻値　　なし

11.8.3. Log

- `void LogStart();`
  ログ取得開始

  引数　　　なし

  戻値　　　なし

- `void LogStop();`
  ログ取得終了

  引数　　　なし

  戻値　　　なし

- `unsigned char isLogRecoding();`
  ログ取得中か確認

  引数　　　なし

  戻値　　　0：停止中　　　1：取得中

- `unsigned char isLogEnable();`
  ログモードが有効か確認

  引数　　　なし

  戻値　　　0：無効 1：有効

- `void setLogEnable();`
  ログモードを有効に設定

  引数　　　なし

  戻値　　　なし

- `void setLogDisable();`
  ログモードを無効に設定

  引数　　　なし

  戻値　　　なし

- `void setLogFlag(unsigned char * val , unsigned short size);`
  ログフラグを設定

  引数　　　*val　設定する変数をメモリマップ変数を（memmap.values.BODY_ANGLE）な形式で指定。

  　　　　　Size　変数のバイト数

  戻値　　　なし

- `void clearLogFlagAll();`

  ログフラグをクリア

  引数　　なし

  戻値　　なし

11.8.4. ADC VS-IX008B

- `unsigned short getIX008(unsigned char num);`

  VS-IX008 から ADC データ取得

  引数　　num　　ポート番号　　[0～7]

  戻値　　12bitADC データ　　[0～4095]

11.8.5. UART

- `void UARTInit(unsigned long baudrate, unsigned char parity , unsigned char stop);`

  UART 初期設定

  引数　　baudrate　　ボーレートを設定　　[列挙子 eBoardRate から選択]

  　　　　parity　　パリティ設定　　[列挙子 eParity から選択]

  　　　　stop　　ストップビット設定　　[1or2]

  戻値　　なし

- `void UARTSendByte(unsigned char data);`

  1 バイト送信

  引数　　data　　バイトデータ

  戻値　　なし

- `unsigned char UARTReadByte(unsigned char *data);`

  1 バイト送信

  引数　　*data　　受信データ格納用変数のポインタ

  戻値　　0：受信データなし　　1：受信データあり

- `void UARTSend(unsigned char *BufferPtr, unsigned long Length);`

  複数バイト送信

  引数　　*BufferPtr　　送信データ配列のポインタ

  　　　　Length　　送信データ配列の長さ

  戻値　　なし

- `void UARTTrace(char *BufferPtr);`
  トレース送信（CR+LF コードを自動負荷）

引数	*BufferPtr	送信文字列のポインタ
戻値	なし	
備考	文字列最後の'¥0'までを自動送信し、改行コードを最後に送信	

11.8.6. モータ制御

- `void SetMotorPwm(int m1, int m2);`
  モータへの PWM 指令値設定

引数	m1,m2	モータへの PWM 指令（-1000～1000）
戻値	なし	

11.9. ゲイン・係数

各ゲイン・係数は以下の動作に関係しています。

A) 本体の制御ゲイン（BODY）

本体の角度・角速度の制御に関するフィードバックゲインです。

角度・角速度を同時に増減させたいときに使用します。

B) 本体角度ゲイン（BODY ANGLE）

本体の傾き（角度）に対するフィードバックゲインです。

本体が傾いた時に転倒しないようにするために、フィードバックを行います。

C) 本体角速度（BODY ANGURUER SPD）

本体が傾くときの速度（角速度）に対するフィードバックゲインです。

本体角度のゲインによる発振を抑える働きがあります。

D) ホイールの制御ゲイン（WHEEL）

ホイールの回転角度・角速度の制御に関するフィードバックゲインです。

角度・角速度を同時に増減させたいときに使用します。

E) ホイール角度ゲイン（WHEEL ANGLE）

ホイールの回転（角度）に対するフィードバックゲインです。

ホイールが回転し続け移動しないようにするためのフィードバックを行います。

F)　ホイール角速度ゲイン（WHEEL ANGURUER SPD）

ホイールが回転する際の速度（角速度）に対するフィードバックゲインです。

ホイール角度のゲインによる発振を抑える働きがあります。

G)　ホイール相関ゲイン（WHEEL CORRELATION）

左右のホイール間の偏差を補正するためのフィードバックゲインです。

角度・角速度を同時に増減させたいときに使用します。

H)　ホイール相関角度ゲイン（WHEEL CORRELATION ANGLE）

左右のホイールの回転（角度）の偏差に対するフィードバックゲインです。

左右のホイールの回転がバラバラだと本体が回転してしまうので、それを抑えるためにフィードバックを行います。

I)　ホイール相関角速度ゲイン（WHEEL CORRELATIO NANGURUER SPD）

左右のホイールが回転する際の速度（角速度）の偏差に対するフィードバックゲインです。

ホイール相関角度のゲインによる発振を抑える働きがあります。

J)　電流制御比例ゲイン（CURRENT P）

K)　電流制御積分ゲイン（CURRENT I）

DC モータの電流制御のフィードバックゲインです。

P ゲインは電流指令値の偏差、I ゲインは偏差の積分値に対するフィードバックゲインです。

L)　ジャイロセンサ HPF 係数（GYRO HPF）

ジャイロセンサのドリフト抑制のために以下の様な簡易な HPF を実装されており、そのカットオフ周波数を変更するための係数です。

$$\mathrm{Angle(t)} = \mathrm{G_{HPF}} \times \mathrm{Angle(t-1)} + \frac{(\mathrm{Aspd(t)} + \mathrm{Apsd(t-1)} \times 0.5)}{\mathrm{dt}}$$

０～１の間で変更でき、値を下げるとカットオフ周波数が上がり、値を１にするとカットオフ周波数が 0 Hz（HPF が OFF）になります。

M) オプショナルゲイン（OPTION　1~6）

任意のゲインを FlashROM に保存できます。

C 言語で制御プログラムを作成後、そのプログラムで使用するゲイン・係数をバランサー2 プログラマから FlashROM に保存することが可能です。

※Microsoft、Visual Studio は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

※Windows の正式名称は、Microsoft Windows Operating System です。

※その他、記載されている会社名、製品名は、各社の登録商標または商標です。